

N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM
MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT
CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED
IN THE INTEREST OF MAKING AVAILABLE AS MUCH
INFORMATION AS POSSIBLE

"Made available under NASA sponsorship
in the interest of early and wide dis-
semination of Earth Resources Survey
Program information and without liability
for any use made thereof."

DRg
8.0-10202

JSC-14554

NASA CR-

160598

"AS-BUILT" SPECIFICATION FOR CCIT7
PROCESSOR PROGRAM

Job Orders 73-783 and 71-475

(TIRF 78-0022)

(E80-10209) AS-BUILT SPECIFICATION FOR
CCIT7 PROCESSOR PROGRAM (Lockheed
Electronics Co.) 64 p HC A04/MF A01

N80-23788

CSCL 05B

G3/43 UNCLAS
00209

Prepared By

Lockheed Electronics Company, Inc.
Systems and Services Division
Houston, Texas

Contract NAS 9-15200

For

EARTH OBSERVATIONS DIVISION
SPACE AND LIFE SCIENCES DIRECTORATE



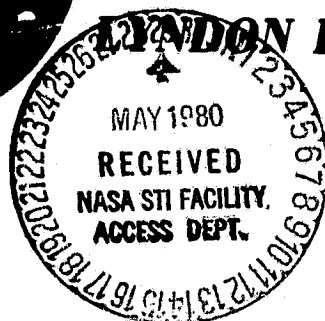
National Aeronautics and Space Administration

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

November 1978

LEC-12518



"AS-BUILT" SPECIFICATION FOR CCIT7
PROCESSOR PROGRAM

Job Orders 73-783 and 71-473
(TIRF 78-0022)

PREPARED BY

W. P. White

APPROVED BY

for James A. Wilkinson
P. L. Krumm, Acting Supervisor
Scientific Applications Software Section

E. M. Hsu
E. M. Hsu, Supervisor
Accuracy Assessment Section

Prepared By

Lockheed Electronics Company, Inc.

For

Earth Observations Division

Space and Life Sciences Directorate

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

November 1978

1. Report No. JSC-14554		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle "As-Built" Specification for CCIT7 Processor Program				5. Report Date November 1978	
				6. Performing Organization Code	
7. Author(s) W. P. White Lockheed Electronics Company, Inc.				8. Performing Organization Report No. LEC-12518	
9. Performing Organization Name and Address Lockheed Electronics Company, Inc. 1830 NASA Road 1 Houston, Texas 77058				10. Work Unit No.	
				11. Contract or Grant No. NAS 9-15200	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center Houston, Texas 77058 Technical Monitor: J. D. Erickson				13. Type of Report and Period Covered Technical Report	
				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract The program CCIT7 is a utility module of the Accuracy Assessment Software System of the Large Area Crop Inventory Experiment. This program accesses data originating on the Classification and Mensuration Subsystem/Crop Assessment Subsystem interface tapes (CCIT's) of the Earth Resources Interactive Processing System (version 7). The data items needed for subsequent Accuracy Assessment processing are written into three disk files. The data extracted consist of a summary of the classification stratified areal estimate, cluster-dot match, and analyst-labeled dots (types 1 and 2).					
17. Key Words (Suggested by Author(s)) Accuracy Assessment CCIT				18. Distribution Statement	
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 65	
				22. Price*	

*For sale by the National Technical Information Service, Springfield, Virginia 22161

JSC Form 1424 (Rev Nov 75)

NASA — JSC

PRECEDING PAGE BLANK NOT FILMED

CONTENTS

Section	Page
ABBREVIATIONS.	vii
1. SCOPE.	1-1
2. APPLICABLE DOCUMENTS	2-1
3. SYSTEM DESCRIPTION	3-1
3.1 <u>HARDWARE DESCRIPTION</u>	3-1
3.2 <u>MODULE DESCRIPTION</u>	3-1
3.3 <u>SOFTWARE DESCRIPTION</u>	3-6
3.3.1 MODULE CCIT7	3-7
3.3.2 SUBROUTINE READH	3-13
3.3.3 SUBROUTINE READRC.	3-17
3.3.4 SUBROUTINE HEADER.	3-20
3.3.5 SUBROUTINE BIASC7.	3-24
3.3.6 SUBROUTINE CLUST7.	3-28
3.3.7 SUBROUTINE RITE7	3-32
3.3.8 SUBROUTINE TURNON.	3-35
3.3.9 SUBROUTINE DOTS7	3-38
3.3.10 SUBROUTINE STCOD7.	3-44
3.3.11 SUBROUTINE PRINT7.	3-47
4. OPERATIONS	4-1
4.1 <u>OPERATORS GUIDE</u>	4-1
4.1.1 HARDWARE CONFIGURATION	4-1
4.1.2 PROGRAM EXECUTION.	4-1

Section	Page
4.2 <u>USERS GUIDE</u>	4-2
4.3 <u>MAINTENANCE DOCUMENTATION</u>	4-2
APPENDIX — FORMAT FOR .CLO FILE.	A-1

TABLES

Table	Page
1 TASK-BUILDER COMMAND FILE FOR CCIT7 PROCESSOR PROGRAM	3-6
2 BATCH RUN DECK SETUP.	4-3

FIGURES

Figure		Page
1	Data flow of the CCIT7 processor program.	3-2
2	Functional flow of the CCIT7 processor program.	3-3
3	Flow diagram for the CCIT7 processor program.	3-9
4	Listing for the CCIT7 processor program	3-11
5	Flow diagram for subroutine READH	3-15
6	Listing for subroutine READH.	3-16
7	Flow diagram for subroutine READRC.	3-18
8	Listing for subroutine READRC	3-19
9	Flow diagram for subroutine HEADER.	3-22
10	Listing for subroutine HEADER	3-23
11	Flow diagram for subroutine BIASC7.	3-25
12	Listing for subroutine BIASC7	3-26
13	Flow diagram for subroutine CLUST7.	3-30
14	Listing for subroutine CLUST7	3-31
15	Flow diagram for subroutine RITE7	3-33
16	Listing for subroutine RITE7.	3-34
17	Flow diagram for subroutine TURNON.	3-36
18	Listing for subroutine TURNON	3-37
19	Flow diagram for subroutine DOTS7	3-40
20	Listing for subroutine DOTS7.	3-42
21	Flow diagram for subroutine STCOD7.	3-45
22	Listing for subroutine STCOD7	3-46
23	Flow diagram for subroutine PRINT7.	3-48
24	Listing for subroutine PRINT7	3-49

ABBREVIATIONS

AA	Accuracy Assessment
CAMS	Classification and Mensuration Subsystem
CAS	Crop Assessment Subsystem
CCIT	CAMS/CAS interface tape
DEC	Digital Equipment Corporation
DPR	Data processing request
DTL	Data Techniques Laboratory
DTRM	Data terminal
EOD	Earth Observations Division
ERIPS	Earth Resources Interactive Processing System
LACIE	Large Area Crop Inventory Experiment
Pixel	Picture element
SAE	Stratified areal estimate
TIRF	Transmittal Information Request Form
UIC	User identification code

1. SCOPE

This document specifies the detailed design for a software module called CCIT7, which will manipulate and extract data from Accuracy Assessment (AA) data base files. These files are derived from Large Area Crop Inventory Experiment (LACIE), version 7, Classification and Mensuration Subsystem/Crop Assessment Subsystem (CAMS/CAS) interface tapes (CCIT's). The data extracted are output into three new data base files for direct input to AA analytical programs.

2. APPLICABLE DOCUMENTS

The following documents, of the exact issue shown, form parts of the specification to the extent specified herein.

- a. "As-Built" Design Specification for PDP 11/45 Accuracy Assessment System Using Disk Data File. JSC-13893 (LEC-11881), February 1978 (and references therein).
- b. Implementation of CCIT6A Processor Program. Transmittal Information Request Form (TIRF) 78-0022, May 11, 1978.
- c. CAM/CAS Interface Tape Interface Control Document. LACIE-C00708, revision A (JSC-09866), July 1976.
- d. Classification and Mensuration Subsystem (CAMS) Requirements. LACIE-C00200, volume II, revision D (JSC-11330), August 1977.
- e. "As-Built" Design Specification for CCIT6A Processor Program. JSC-14368 (LEC-12303), August 1978.

3. SYSTEM DESCRIPTION

The CCIT7 processor module accomplishes the data manipulations shown in figure 1. Basically, the CCIT data for a particular segment number, SSSS, and classification date, YYDDD, contained in file SSSSYDDDD.CC0 are processed to obtain three output files required as input to existing or planned AA programs. The SSSSYDDDD.CLO file contains data needed for future programs. The SSSSYDDDD.AI1 and SSSSYDDDD.AI2 files are required for input to existing modules SPATL and MLTCRP.

3.1 HARDWARE DESCRIPTION

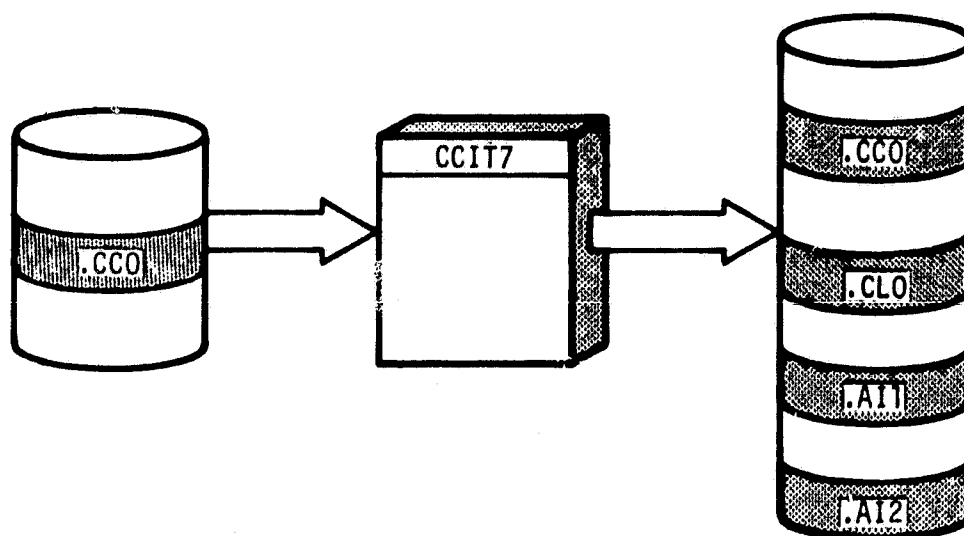
The PDP 11/45, with the following peripherals, is required.

- a. Card reader or user terminal
- b. Line printer
- c. Two disk units

3.2 MODULE DESCRIPTION

The CCIT7 module is implemented on the PDP 11/45 for time-sharing or background processing of CCIT data files into three output data files: an unformatted file of character data and two formatted files of analyst-labeled dots. See the functional flow diagram (fig. 2). The CCIT7 module represents a major revision of the CCIT6A module described in reference e (section 2).

The LACIE CCIT is a universal nonimaging tape containing extensive statistical and ancillary data for a series of Earth Resources Interactive Processing System (ERIPS) runs. Using the AA CCIT program, all data for a relevant segment are transferred to a Files-11 disk file named SSSSYDDDD.CC0,..., where SSSS is the segment number, YY is the year, and DDD is the day of the year. This .CC0 file contains three 80-byte header records and a large number (>20) of 720-byte data records.



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 1.— Data flow of the CCIT7 processor program.

~~3-2~~
4

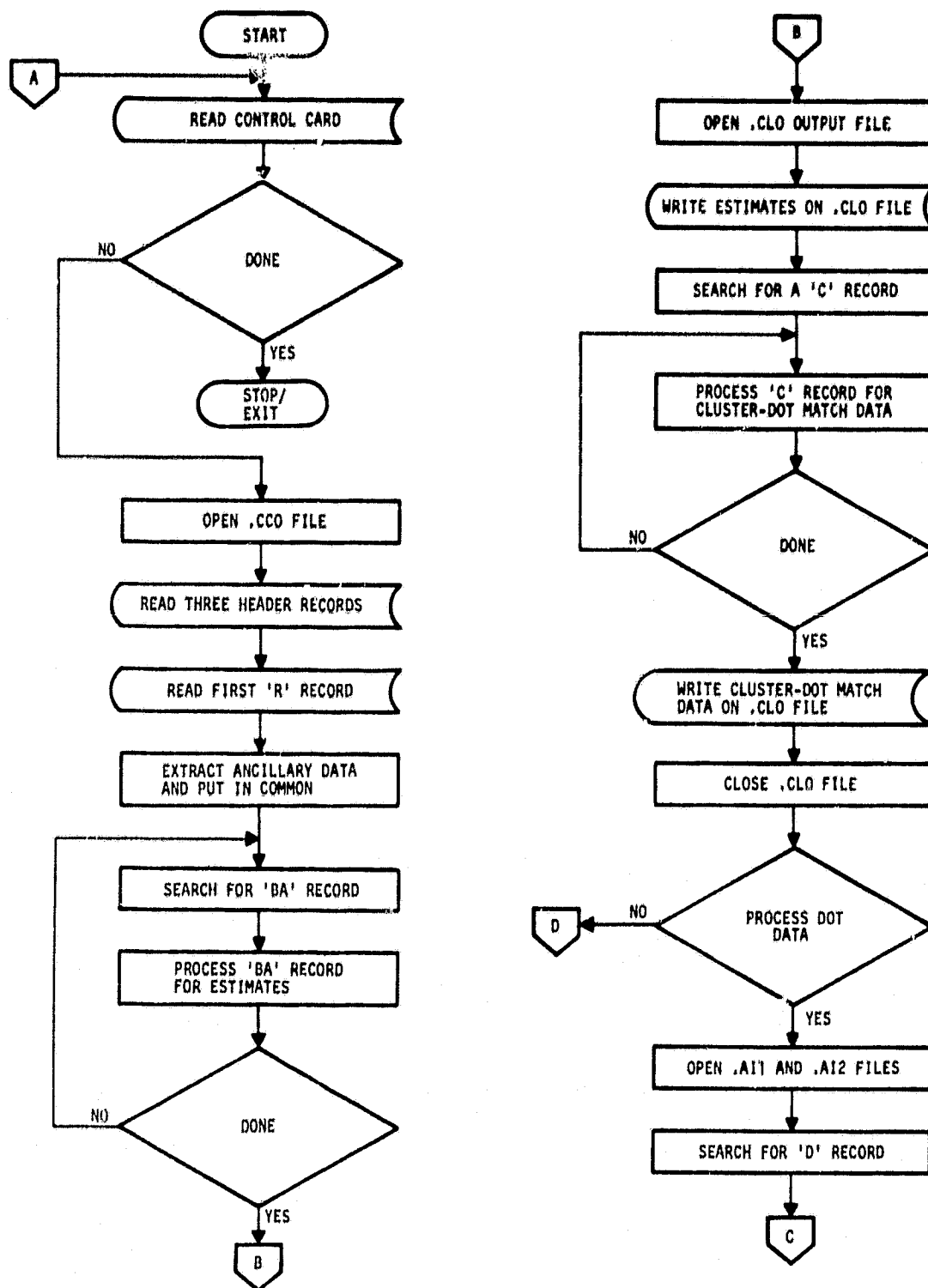


Figure 2.— Functional flow of the CCIT7 processor program.

3-3
5

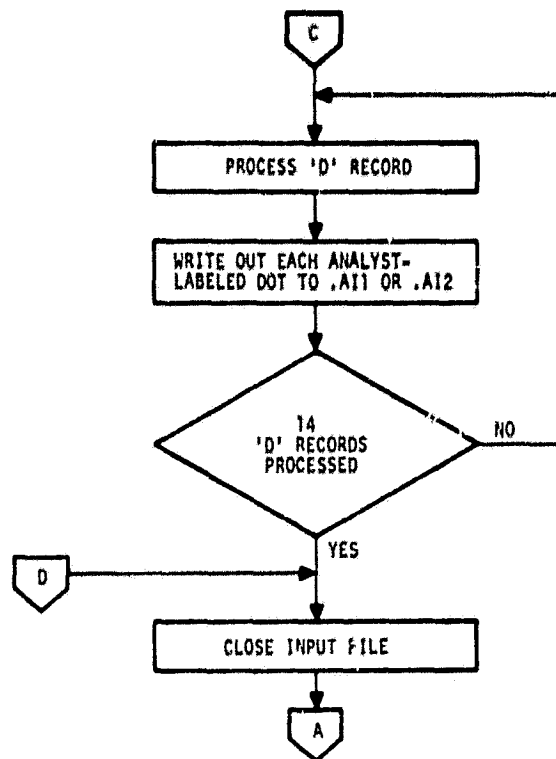


Figure 2.— Concluded.

3-4
6

The first step of the process is to read the name of the input .CCO file and open this file for reading. Then the three CCIT header records are read and ignored. The next record (720 bytes) is read, checked to verify that it is a recognition (R) record, and processed to extract the data processing request (DPR) number and acquisition dates used in the classification.

Next a 'BA' record is searched for and processed. The program extracts classification data on class picture element (pixel) populations, the ERIPS estimate, the stratified areal estimate (SAE), and the variance for each class. An output file SSSSYDDDD.CLO (where SSSS is the segment number and YYDDD is the classification date derived from the DPR number) is opened. The first record of the .CLO file contains an integer word giving the number of crop classes for the classification. The estimate data are written as the second record. Sixteen bytes are required for each class; a LACIE version 7 CCIT can contain data on up to 26 classes. A line printer report is output after the disk files are written.

In the next step a 'C' record is searched for and processed. The total number of clusters (Q) and cluster-dot match data are extracted as Q-groups of 12 characters. (Generally, there are more than 15 clusters, so some of these data appear in additional 'C' records.) When all the cluster-dot data are assembled into a buffer, the number of clusters is written out as the third record of file SSSSYDDDD.CLO, and the match data are written as the second record of this file. A line printer report is then output, and the SSSSYDDDD.CLO file is closed.

Then the program tests to determine if analyst-labeled dot output files are required; this is the default condition. If this condition exists, the output files SSSSYDDDD.AI1 and SSSSYDDDD.AI2 are opened, and a search is made for the first dot record. There are 14 dot records listing all 209 dots. The program examines each dot to determine if it has been labeled by the analyst. If so, it is written out (line, sample, and label) to the proper file, depending on the dot type (1 or 2). The first dot record in

each output file also contains ancillary information on the segment (number and state code), classification date, acquisition dates, data terminal (DTRM) tape number, and type of label.

When all dots have been processed, the SSSSYDDDD.AI1 and SSSSYDDDD.AI2 output files and the input file are closed. No report is output detailing the .AI file dot data because these formatted files are easily examined using the Digital Equipment Corporation (DEC) PDP-11 utility program PIP.

3.3 SOFTWARE DESCRIPTION

The CCIT7 processor program consists of 11 user-supplied routines: CCIT7 (main program), READH, READRC, HEADER, BIASC7, CLUST7, RITE7, TURNON, DOT7, STCOD7, and PRINT7. The program makes use of a card-image-formatted file, CCIT7.DAT, for program control and the line printer and user disk for output. The following sections provide a detailed description of each of the 11 routines. The recommended task-build command file (CCIT7.CMD), used to create the load module (CCIT7.TSK), is given in table 1.

TABLE 1.-- TASK-BUILDER COMMAND FILE FOR CCIT7 PROCESSOR PROGRAM

```
CCIT7,LP:/SH=CCIT7,READH,READRC,HEADER,BIASC7,CLUST7,  
RITE7,TURNON,DOT7,STCOD7,PRINT7  
/  
FMTBUF=132  
UNITS=6  
ACTFIL=6  
ASG=SY:1  
ASG=SY:2  
ASG=SY:3  
ASG=SY:5  
ASG=LP:6  
PRI=50  
//
```

For simplicity, the definition of arrays carried in COMMON blocks, the definition of COMMON blocks, and the description of COMMON blocks are not repeated for each routine. Instead, each of these elements is described in the

routine of origin. Reference to the Interface subsections and to the compiler listings of each routine provides sufficient information to follow the data flow throughout the program.

3.3.1 MODULE CCIT7

3.3.1.1 Linkage

The CCIT7 program is the main program. It calls user subroutines READH, READRC, HEADER, BIASC7, CLUST7, DOT7, and PRINT7. Subroutines BIASC7, DOT7, and PRINT7 are called using multiple entry points.

3.3.1.2 Interface

Most communication with the user routines is handled via COMMON blocks. A single integer parameter is passed on call to READH, which indicates the number of CCIT header records to be read. A single integer parameter Q is passed to CLUST7 as a flag.

3.3.1.2.1 COMMON Block BUF

BUF contains a 720-byte array, A, which is used to hold one CCIT logical record for processing.

3.3.1.2.2 COMMON Block FNAME

FNAME contains a 24-byte array, FILNAM, and an integer variable, SKIP. FILNAM contains the input file name read from CCIT7.DAT. The value of SKIP determines whether the dot records are to be processed. If SKIP is nonzero, the dots are not processed.

3.3.1.2.3 COMMON Block B7

B7 contains the number of categories, NCAT, for the classification.

3.3.1.2.4 COMMON Block CLUSTR

CLUSTR contains a 60-byte by 12-byte array, CNAME, and an integer variable CNUM. CLUSTR provides an interface between subroutine CLUST7 and subroutine RITE7.

3.3.1.3 Input

The CCIT7 program receives all input CCIT data via subroutine READRC. Control data are provided directly by reading card images from file CCIT7.DAT.

3.3.1.4 Output

The CCIT7 program provides all output via subroutines PRINT7, RITE7, BIASC7, DOTS7, and CLUST7.

3.3.1.5 Storage

The CCIT7 program requires 1078 words of storage.

3.3.1.6 Description

The CCIT7 routine provides the control function for the program. Flow is controlled via tests on the first bytes (descriptive characters) of each logical record in the CCIT input file and by counts based upon the required number of records of a given type.

3.3.1.7 Flow Chart

The flow chart for CCIT7 is given in figure 3.

3.3.1.8 Listing

The listing for this subroutine is given in figure 4.

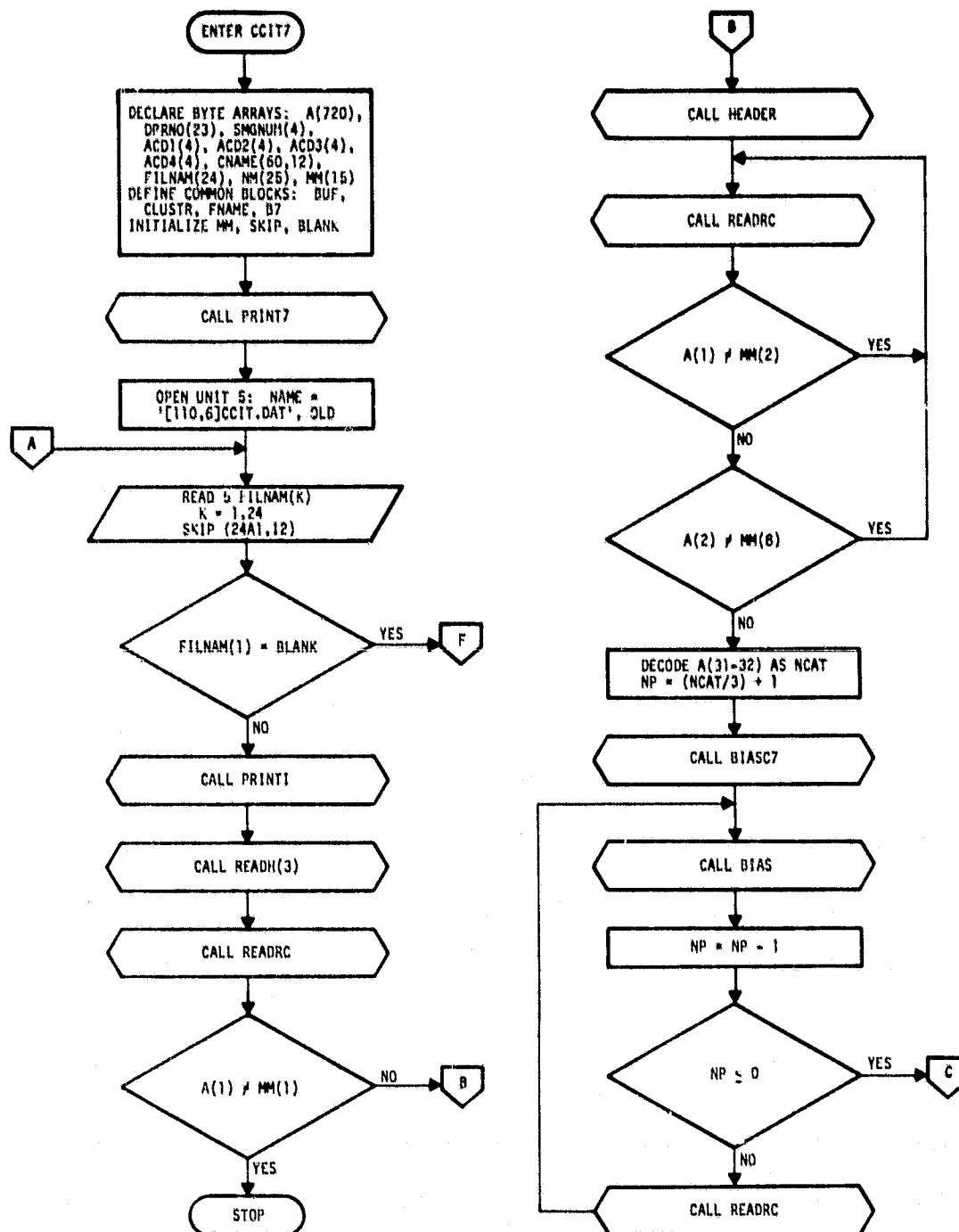


Figure 3.— Flow diagram for the CCIT7 processor program.

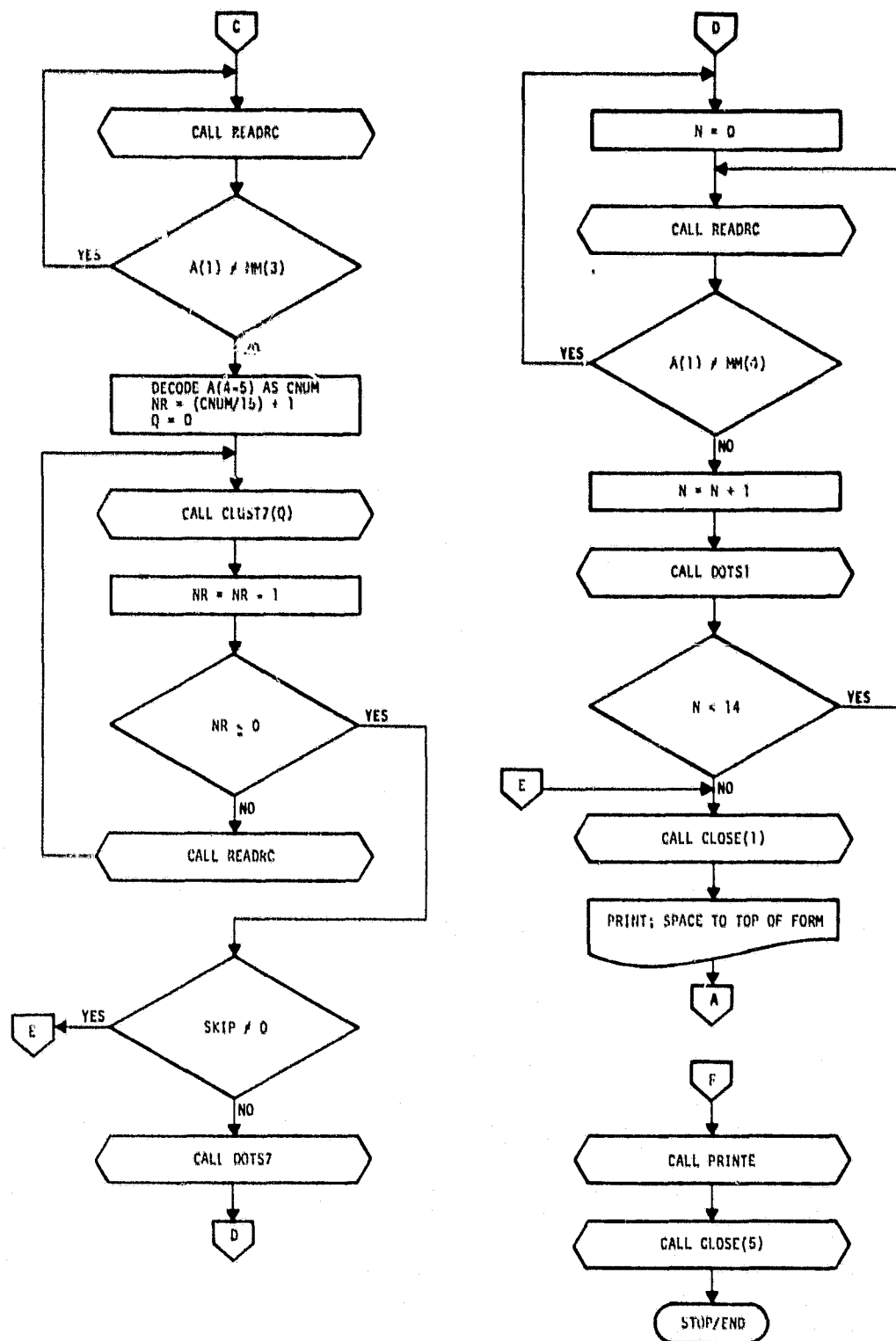


Figure 3.— Concluded.

3-10
12

```

0001 PROGRAM CCIT7
0002 IMPLICIT INTEGER(A-Z)
0003 SAVE A(25),DPNR(25),SHRNR(4),ACR(4),JCN(4),NCR(4),
1 CHAME(60,12),FILNAM(24),NP(25),PH(15),BLANK
C.....
C MAIN OF CCIT7 PROCESSOR - READ A CARD IMAGE FROM [118,0]CCIT7.DAT
C WHICH GIVES THE FILE NAME OF THE LACIES AS CCIT7 FILE ON DISC
C IN THE FORM: [131,7]SSSYDDO.CGO
C A(20) IS THE BUFFER WHICH CONTAINS ONE CCIT7 DATA RECORD
C.....
0004 COMMON/BUFA
0005 COMMON/B7/NCAT
0006 COMMON/CLUSTER/CRNR,CNRN
0007 COMMON/FNAME/FILNAM,SKIP
0008 DATA H(1MR,1MG,1MC,1MD,1P,1S,1SH,1MG,10IN) /
0009 DATA SKIP/0/
0010 DATA BLANK/IN /
0011 CALL PRINT7
C
0012 READ INPU FILNAME AND SKIP FACTOR
0013 OPEN UNIT 9, NAME='[110,0]CCIT7.CAT', TYPE='OLD', READONLY,
1 ACCESS='SEQUENTIAL', FORM='UNFORMATTED')
0014 READ(9,10)(FILNAMIK),K=1,24,SKIP
0015 IF(FILNAM(1).EQ.BL.NK) GO TO 200
CALL PRINT1
C
0016 READ 3 HEADER RECORDS OF 80 BYTES EACH
CALL READN(3)
C
0017 READ THE FIRST 'N' RECORD AND PROCESS FOR DPR NUMBER ACQUISITION
C
0018 DATES ETC
CALL READNC
0019 IF(A(1).NE.NH(1)) STOP
CALL HEADER
C
0020 READ AND PROCESS BIAS CORRECTION SUMMARY RECORD
C AS FIRST RECORD OF FILE SSSSYDDO.CGO
0021 CALL READNC
0022 IF(A(1).NE.NH(2)) GO TO 210
0023 IF(A(2).NE.NH(3)) GO TO 210
0024 DECODE(2,100,A(3)) /CAT
NP = (NCAT/3) + 1
0025 CALL BTASC7
0026 CALL BIAS
205
0027 NP = NP + 1
0028 IF(NP.LE.0) GO TO 220
0029 CALL READNC
0030 GO TO 245
C
0031 READ FIRST CLUSTER RECORD
C WHEN DONE SUBROUTINE 'RITE7' IS CALLED WHICH WRITES THE NUMBER
C OF CLUSTERS AND THE CLUSTER-DATA MATCH DATA AS TWO RECORDS
C OF FILE SSSSYDDO.LO
0032 CALL READNC
0033 IF(A(1).NE.NH(3)) GO TO 220
0034 DECODE(2,100,A(4)) /CNRN
NR = (CNRN/15) + 1
0035 C = 0
0036 235 CALL CLUST7(0)
0037 NR = NR + 1
0038 IF(NR.LE.0) GO TO 225

```

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 4.- Listing for the CCIT7 processor program.

```

FORTRAN IV-PLUS V02-51      12143106      17-JUL-78      PAGE 2
CG177.PTN      /TRIPLCKS/MQ

0039      CALL READRC
0040      GO TO 235
0041      C      IF DAY DATA IS NOT NEEDED SKIP TO END OF PROGRAM
0042      C      IF (BKT(ME(0)) GO TO 235
0043      C      INITIALIZE THE DAY OUTPUT FILES AND COUNTERS
0044      C      CALL DBTS7
0045      C      READ DBTS RECORDS
0046      C      N = 0
0047      C      LOOK FOR 'D' (200 DAYS) RECORDS
0048      C      CALL READRC
0049      IF (I(1).NE. 'M'(0)) GO TO 230
0050      N = N + 1
0051      C      PREPARE A 'D' RECORD
0052      CALL DBTS1
0053      IF (N.EQ.14) GO TO 240
0054      C      CLOSE INPUT FILE
0055      CALL CLOSE(1)
0056      PRINT 102
0057      GO TO 200
0058      C      CALL DBTS7
0059      CALL CLOSE(5)
0060      STOP
0061      FORMAT(12)
0062      FORMAT(24A1,12)
0063      FORMAT(3H1)
0064      END

```

Figure 4.— Concluded.

3.3.2 SUBROUTINE READH

3.3.2.1 Linkage

Subroutine READH calls subroutine TURNON.

3.3.2.2 Interface

READH interfaces with TURNON via an integer parameter (passed on call) giving the logical unit number to be opened and via COMMON block NAME containing the name of the file to be opened. COMMON block FNAME (see section 3.3.1.2.2) interfaces CCIT7 with READH. COMMON block BUF provides no true interfacing function for this routine.

3.3.2.2.1 COMMON Block NAME

NAME contains a 25-byte array, NM, which contains the complete name of a file to be opened by subroutine TURNON. NAME also interfaces several subroutines with subroutine PRINT7. The last byte of array NM should contain the null (0) character.

3.3.2.3 Input

Header records from the CCIT input file are input.

3.3.2.4 Output

The only output is a read error message to the line printer.

3.3.2.5 Storage

READH requires 504 words of storage.

3.3.2.6 Description

Subroutine READH spaces past the three 80-byte CCIT header records, and the CCIT file name is written into the NM array. Subroutine TURNON opens the file on unit 1, the three records are read, and READH returns to CCIT7.

3.3.2.7 Flow Chart

The flow diagram for subroutine READH is given in figure 5.

3.3.2.8 Listing

The listing for this subroutine is given in figure 6.

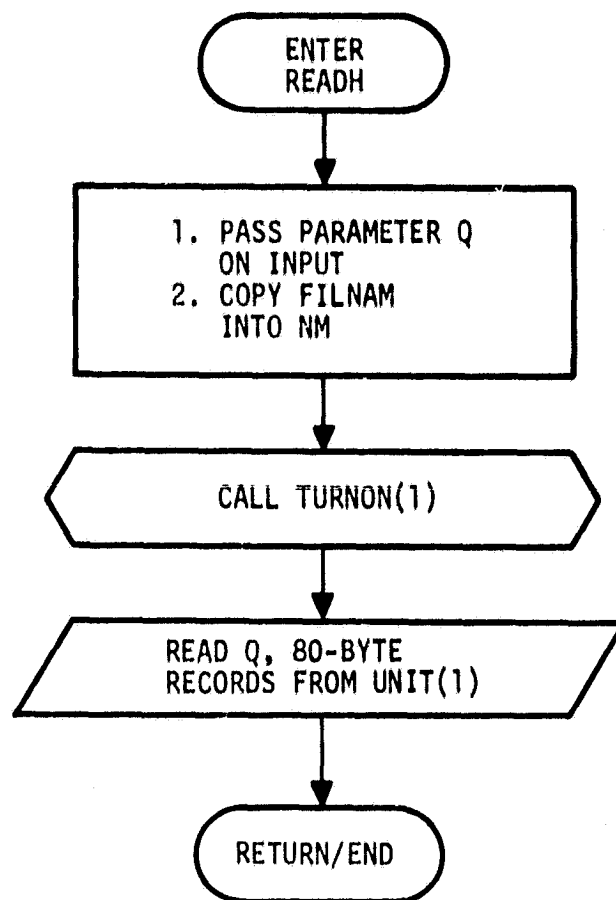


Figure 5.— Flow diagram for subroutine READH.

FORTRAN IV-PLUS V05-51 07144133 05-PAY-78 PAGE 1
 READM,PTN /1010-00-87M

```

0001 SUBROUTINE READMTCJ
0002 IMPLICIT INTEGER(A-Z)
0003 READ EDIT-HEADER-RECORDS FROM DISC FOR FILE FILNAM
0004 BYTE NM(25),FILNAM(24),A(720)
0005 OPEN(UNIT=77,FILE=
0006 COM-LEN/NAME/IN
0007 CORROW/NSIS/FILE/NAME/IN
0008 DO 10 J=1,24
0009 NM(J)=FILNAM(J)
0010 NM(25)=0
0011 CIRC TURN(12)
0012 DO 1 J=1,0
0013 READ(UNIT=77,END=999)NM(1:25)
0014 RETURN
0015 99 PRINT*,NM(1:25)
0016 2*0 FORMAT(1M,' READ ERROR IN READMTCJ ITERATION ',I2,/)
0017 STOP
0018 END
  
```

3-16
 18

ORIGINAL PAGE 1
 OF POOR QUALITY

Figure 6.— Listing for subroutine READM.

3.3.3 SUBROUTINE READRC

3.3.3.1 Linkage

Subroutine READRC is called by CCIT7.

3.3.3.2 Interface

Subroutine READRC interfaces with CCIT7 via COMMON block BUF (see section 3.3.1.2.1).

3.3.3.3 Input

One data record read from the CCIT disk file is input.

3.3.3.4 Output

A read operation error message is output to the line printer.

3.3.3.5 Storage

This subroutine requires 441 words of storage.

3.3.3.6 Description

READRC reads one 720-byte logical data record from the CCIT input file into a buffer array, A.

3.3.3.7 Flow Chart

The flow diagram for subroutine READRC is given in figure 7.

3.3.3.8 Listing

The listing for this subroutine is given in figure 8.

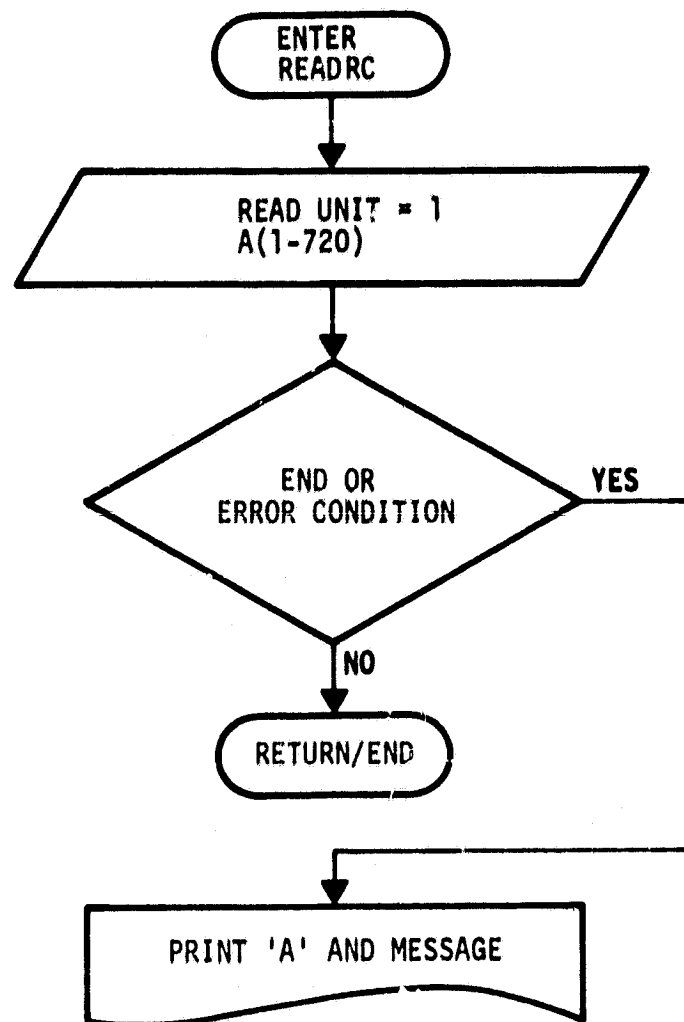


Figure 7.— Flow diagram for subroutine READRC.

```

FORTHAN IV-PLUS V02-51
READRC,IN 07144149 05-PAY-78 PAGE 1
/INTBLRCS/MR

0001 SUBROUTINE READRC
0002 IMPLICIT INTEGER(A-Z)
0003 READ 8 CCIT DATA RECORD OF 320 BYTES INTO THE BUFFER A
0004 BYTE A(720)
0005 COMMON/BUF77
0006 READ(1,ERR=99,END=99)(A(K),K=1,720)
0007 RETURN
0008 90 PRINT 100.4
0009 100 FORTNIN 1, 'ERROR IN READRC+777, BUFFER CONTAINS+777,
0010 01H, 6(120A1)
0011 END

```

Figure 8.— Listing for subroutine READRC.

3.3.4 SUBROUTINE HEADER

3.3.4.1 Linkage

Subroutine HEADER is called by CCIT7.

3.3.4.2 Interface

HEADER interfaces with CCIT7 via COMMON blocks BUF (see section 3.3.1.2.1) and DOTS and interfaces with PRINT7 (entry PRINTH) via COMMON block DOTS.

3.3.4.2.1 COMMON Block DOTS

DOTS contains a 23-byte array (DPRNO) that is used to hold the ERIPS DPR number, four 4-byte arrays (ACD1, ACD2, ACD3, and ACD4) that are used to store the acquisition dates used for the ERIPS run, and a 4-byte array (SMGNUM) that is used to store the LACIE segment number.

3.3.4.3 Input

There is no input to this subroutine.

3.3.4.4 Output

HEADER has no output.

3.3.4.5 Storage

This subroutine requires 445 words of storage.

3.3.4.6 Description

Subroutine HEADER selects byte data from the 'R' record of a CCIT (contained in buffer array A) and stores it into arrays in COMMON block DOTS. The data selected are the LACIE segment number [SMGNUM(1-4)], acquisition dates used for the ERIPS run [ACD1,ACD2,ACD3,ACD4], and ERIPS DPR number [DPRNO(1-23)]. The DPR number is printed in a message via a call to entry PRINTH of PRINT7.

3.3.4.7 Flow Chart

The flow diagram for subroutine HEADER is given in figure 9.

3.3.4.8 Listing

The listing for this subroutine is given in figure 10.

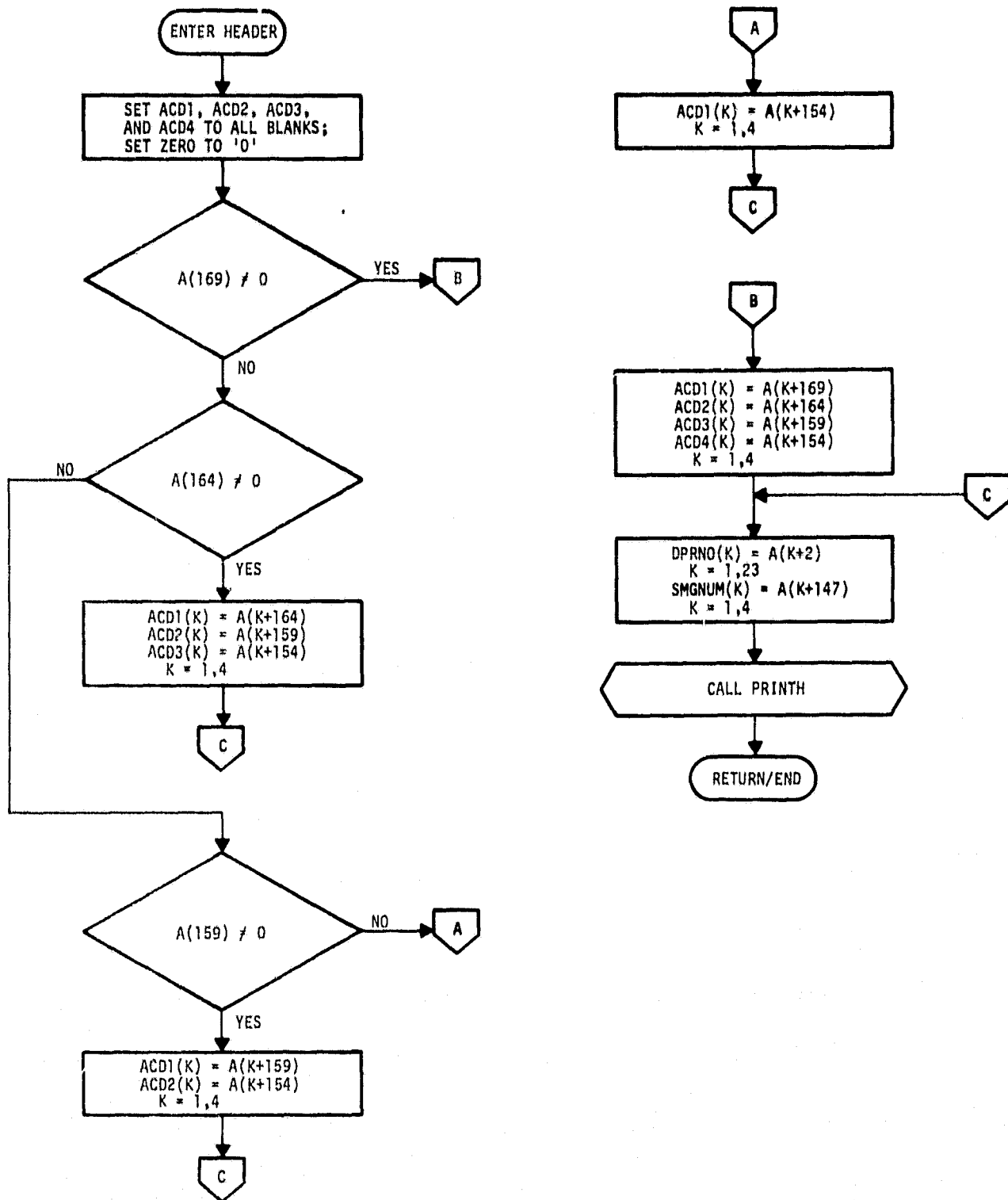


Figure 9.— Flow diagram for subroutine HEADER.


```

0001 SUBROUTINE HEADER
C.....
C THIS SUBROUTINE OBTAINS THE DPR NUMBER AND ACQUISITION
C DATES FROM AN "R" RECORD.
C THE ACQUISITION DATES ARE ORDERED BY LATEST DATE FIRST
C TO COMPARE TO OTHER AA SOFTWARE MODULE CONVENTIONS
C.....
0002 IMPLICIT INTEGER(A-Z)
0003 BYTE A(720),ACD1(4),ACD2(4),ACD3(4),ACD4(4),
0004 COMMON/BUF/1
0005 COMMON/DTS/DPRN,ACD1,ACD2,ACD3,ACD4,SHGNUM
0006 DATA ACD1/401M /ACD2/401M /ACD3/401M /ACD4/401M /
0007 DATA ZER0/180/
C CHECK TO SEE HOW MANY ACQUISITIONS WERE USED
C ASSUMED TO BE IN THE ORDER OF EARLIEST FIRST
C WITH ASCII ZERO FIRST
0008 IF(A(100).NE.ZER0) GO TO 10
0009 IF(A(104).NE.ZER0) GO TO 20
0010 IF(A(150).NE.ZER0) GO TO 30
0011 DO 4 K=1,4
0012 ACD1(K)=A(K*194)
0013 GO TO 40
0014 DO 5 K=1,4
0015 ACD1(K)=A(K*199)
0016 DO 5 K=1,4
0017 ACD2(K)=A(K*194)
0018 GO TO 40
0019 DO 6 K=1,4
0020 ACD1(K)=A(K*104)
0021 ACD2(K)=A(K*199)
0022 DO 6 K=1,4
0023 ACD3(K)=A(K*194)
0024 DO 7 K=1,4
0025 ACD1(K)=A(K*109)
0026 ACD2(K)=A(K*104)
0027 ACD3(K)=A(K*199)
0028 ACD4(K)=A(K*194)
C WRITE THE DPR NUMBER
0029 DO 3 K=1,23
0030 DPRN(K)=A(K*2)
C WRITE SEGMENT NUMBER
0031 DO 1 K=1,4
0032 SHGNUM(K)=A(K*145)
C PRINT MESSAGE ON PRINTER
0033 CALL PRINTM
0034 RETURN
END

```

3-23
35

Figure 10.— Listing for subroutine HEADER.

3.3.5 SUBROUTINE BIASC7

3.3.5.1 Linkage

BIASC7 is called once by the CCIT7 program. It calls subroutine TURNON once. An additional entry point, BIAS, is called one or more times by CCIT7.

3.3.5.2 Interface

BIASC interfaces with CCIT7 via COMMON blocks BUF (see section 3.3.1.2.1) and B7 (see section 3.3.1.2.3), and with TURNON via COMMON block NAME (see section 3.3.2.2.1).

3.3.5.3 Input

There is no input to this subroutine.

3.3.5.4 Output

BIASC7 writes two unformatted records onto unit 3. The data contained in this record are detailed in the appendix.

3.3.5.5 Storage

This subroutine requires 837 words of storage.

3.3.5.6 Description

BIASC7 codes the output file name as SSSSYDDDD.CLO, where SSSS is the segment number and YYDDD is the classification date. Unit 3 is opened for output via a call to subroutine TURNON. Then a record containing the integer NCAT (the number of categories or classes) is written on unit 3, and 12 times NCAT bytes of character data are output as the second record.

3.3.5.7 Flow Chart

The flow diagram for subroutine BIASC7 is given in figure 11.

3.3.5.8 Listing

The listing for this subroutine is given in figure 12.

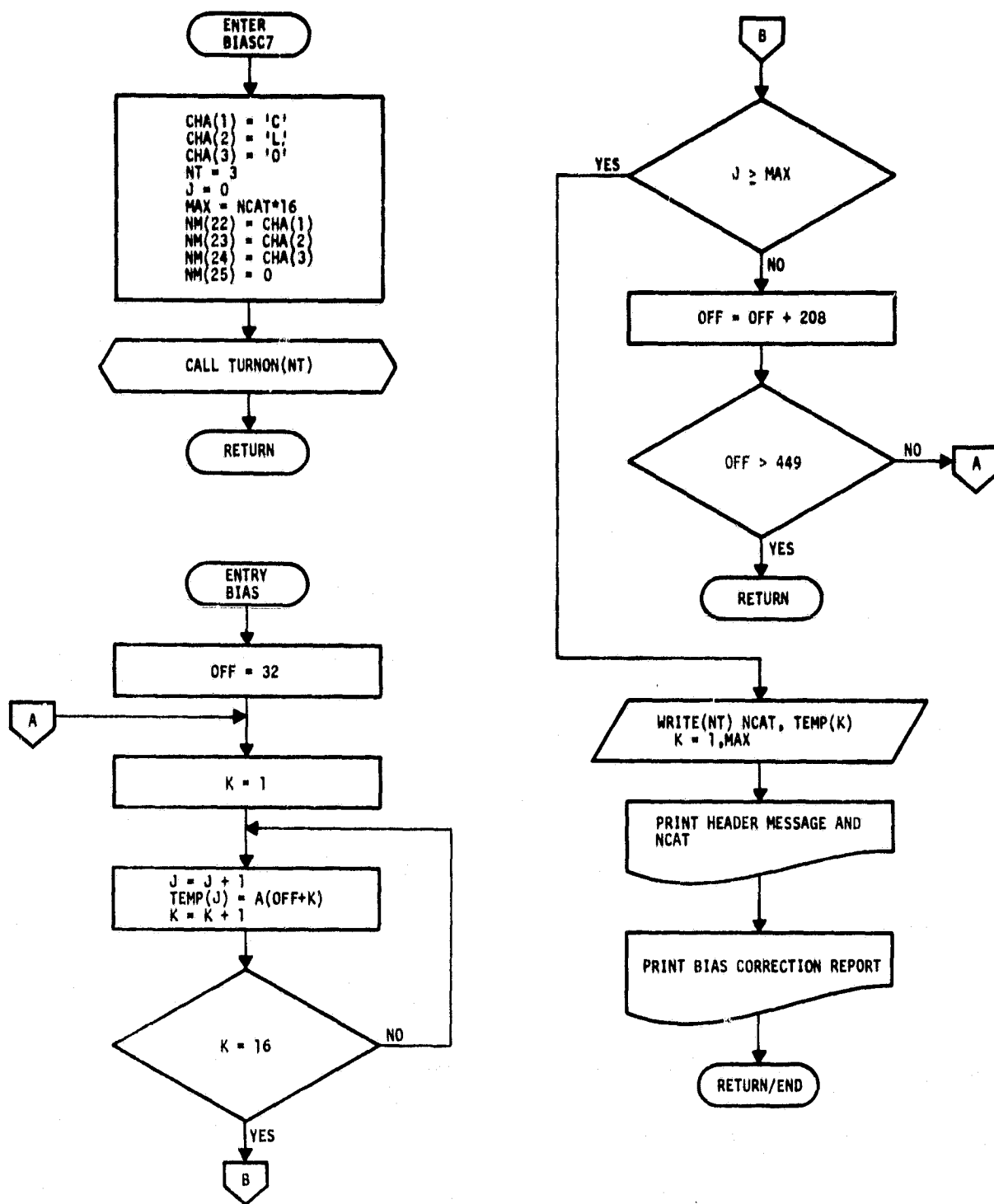


Figure 11.— Flow diagram for subroutine BIASC7.

3-25
27

```

0001 SUBROUTINE BIASCT
0002 IMPLICIT INTEGER(I)
C.....
C USES THE BIAS CORRECTION TECH STRATIFIED AREA
C ESTIMATE DATA FROM A CCIT 'BA' RECORD
C FOR THE FIRST RECORD OF THE
C SSSSYDDJ.CLO OUTPUT FILE, LEAVES FILE OPEN FOR
C WRITING OF THE FIRST RECORD NUMBER 2F CLUSTERS,
C AND THE ALPHANUMERIC C CLUSTER-CBY MATCH TABLE
C BY SUBROUTINE CLUST7
C.....
0003 NAME = 'B77.CIT'
0004 DATE = N(25), SGNUM(4), DPNR(23), ACD1(4), ACD2(4), ACD3(4),
C 1 ACD4(4), ACD5(4), ACD6(4), ACD7(4), ACD8(4), ACD9(4), ACD10(4),
C 11 ACD11(4), ACD12(4), ACD13(4), ACD14(4), ACD15(4), ACD16(4),
C 17 ACD17(4), ACD18(4), ACD19(4), ACD20(4), ACD21(4), ACD22(4), ACD23(4),
C 25 ACD24(4), ACD25(4), ACD26(4), ACD27(4), ACD28(4), ACD29(4), ACD30(4),
C 31 ACD31(4), ACD32(4), ACD33(4), ACD34(4), ACD35(4), ACD36(4), ACD37(4),
C 39 ACD38(4), ACD39(4), ACD40(4), ACD41(4), ACD42(4), ACD43(4), ACD44(4),
C 47 ACD45(4), ACD46(4), ACD47(4), ACD48(4), ACD49(4), ACD50(4), ACD51(4),
C 55 ACD52(4), ACD53(4), ACD54(4), ACD55(4), ACD56(4), ACD57(4), ACD58(4),
C 59 ACD59(4), ACD60(4), ACD61(4), ACD62(4), ACD63(4), ACD64(4), ACD65(4),
C 67 ACD66(4), ACD67(4), ACD68(4), ACD69(4), ACD70(4), ACD71(4), ACD72(4),
C 75 ACD73(4), ACD74(4), ACD75(4), ACD76(4), ACD77(4), ACD78(4), ACD79(4),
C 83 ACD80(4), ACD81(4), ACD82(4), ACD83(4), ACD84(4), ACD85(4), ACD86(4),
C 87 ACD87(4), ACD88(4), ACD89(4), ACD90(4), ACD91(4), ACD92(4), ACD93(4),
C 95 ACD94(4), ACD95(4), ACD96(4), ACD97(4), ACD98(4), ACD99(4), ACD100(4),
C 107 ACD101(4), ACD102(4), ACD103(4), ACD104(4), ACD105(4), ACD106(4),
C 111 ACD107(4), ACD108(4), ACD109(4), ACD110(4), ACD111(4), ACD112(4),
C 115 ACD113(4), ACD114(4), ACD115(4), ACD116(4), ACD117(4), ACD118(4),
C 119 ACD119(4), ACD120(4), ACD121(4), ACD122(4), ACD123(4), ACD124(4),
C 127 ACD125(4), ACD126(4), ACD127(4), ACD128(4), ACD129(4), ACD130(4),
C 135 ACD131(4), ACD132(4), ACD133(4), ACD134(4), ACD135(4), ACD136(4),
C 139 ACD137(4), ACD138(4), ACD139(4), ACD140(4), ACD141(4), ACD142(4),
C 145 ACD143(4), ACD144(4), ACD145(4), ACD146(4), ACD147(4), ACD148(4),
C 151 ACD149(4), ACD150(4), ACD151(4), ACD152(4), ACD153(4), ACD154(4),
C 157 ACD155(4), ACD156(4), ACD157(4), ACD158(4), ACD159(4), ACD160(4),
C 163 ACD161(4), ACD162(4), ACD163(4), ACD164(4), ACD165(4), ACD166(4),
C 167 ACD167(4), ACD168(4), ACD169(4), ACD170(4), ACD171(4), ACD172(4),
C 175 ACD173(4), ACD174(4), ACD175(4), ACD176(4), ACD177(4), ACD178(4),
C 183 ACD179(4), ACD180(4), ACD181(4), ACD182(4), ACD183(4), ACD184(4),
C 187 ACD185(4), ACD186(4), ACD187(4), ACD188(4), ACD189(4), ACD190(4),
C 195 ACD191(4), ACD192(4), ACD193(4), ACD194(4), ACD195(4), ACD196(4),
C 197 ACD197(4), ACD198(4), ACD199(4), ACD200(4), ACD201(4), ACD202(4),
C 207 ACD203(4), ACD204(4), ACD205(4), ACD206(4), ACD207(4), ACD208(4),
C 211 ACD209(4), ACD210(4), ACD211(4), ACD212(4), ACD213(4), ACD214(4),
C 215 ACD215(4), ACD216(4), ACD217(4), ACD218(4), ACD219(4), ACD220(4),
C 227 ACD221(4), ACD222(4), ACD223(4), ACD224(4), ACD225(4), ACD226(4),
C 231 ACD227(4), ACD228(4), ACD229(4), ACD230(4), ACD231(4), ACD232(4),
C 235 ACD233(4), ACD234(4), ACD235(4), ACD236(4), ACD237(4), ACD238(4),
C 239 ACD239(4), ACD240(4), ACD241(4), ACD242(4), ACD243(4), ACD244(4),
C 247 ACD245(4), ACD246(4), ACD247(4), ACD248(4), ACD249(4), ACD250(4),
C 251 ACD251(4), ACD252(4), ACD253(4), ACD254(4), ACD255(4), ACD256(4),
C 257 ACD257(4), ACD258(4), ACD259(4), ACD260(4), ACD261(4), ACD262(4),
C 263 ACD263(4), ACD264(4), ACD265(4), ACD266(4), ACD267(4), ACD268(4),
C 267 ACD269(4), ACD270(4), ACD271(4), ACD272(4), ACD273(4), ACD274(4),
C 275 ACD275(4), ACD276(4), ACD277(4), ACD278(4), ACD279(4), ACD280(4),
C 283 ACD281(4), ACD282(4), ACD283(4), ACD284(4), ACD285(4), ACD286(4),
C 287 ACD287(4), ACD288(4), ACD289(4), ACD290(4), ACD291(4), ACD292(4),
C 295 ACD293(4), ACD294(4), ACD295(4), ACD296(4), ACD297(4), ACD298(4),
C 297 ACD299(4), ACD300(4), ACD301(4), ACD302(4), ACD303(4), ACD304(4),
C 307 ACD305(4), ACD306(4), ACD307(4), ACD308(4), ACD309(4), ACD310(4),
C 311 ACD311(4), ACD312(4), ACD313(4), ACD314(4), ACD315(4), ACD316(4),
C 315 ACD317(4), ACD318(4), ACD319(4), ACD320(4), ACD321(4), ACD322(4),
C 319 ACD323(4), ACD324(4), ACD325(4), ACD326(4), ACD327(4), ACD328(4),
C 323 ACD329(4), ACD330(4), ACD331(4), ACD332(4), ACD333(4), ACD334(4),
C 327 ACD335(4), ACD336(4), ACD337(4), ACD338(4), ACD339(4), ACD340(4),
C 331 ACD341(4), ACD342(4), ACD343(4), ACD344(4), ACD345(4), ACD346(4),
C 335 ACD347(4), ACD348(4), ACD349(4), ACD350(4), ACD351(4), ACD352(4),
C 339 ACD353(4), ACD354(4), ACD355(4), ACD356(4), ACD357(4), ACD358(4),
C 343 ACD359(4), ACD360(4), ACD361(4), ACD362(4), ACD363(4), ACD364(4),
C 347 ACD365(4), ACD366(4), ACD367(4), ACD368(4), ACD369(4), ACD370(4),
C 351 ACD371(4), ACD372(4), ACD373(4), ACD374(4), ACD375(4), ACD376(4),
C 355 ACD377(4), ACD378(4), ACD379(4), ACD380(4), ACD381(4), ACD382(4),
C 359 ACD383(4), ACD384(4), ACD385(4), ACD386(4), ACD387(4), ACD388(4),
C 363 ACD389(4), ACD390(4), ACD391(4), ACD392(4), ACD393(4), ACD394(4),
C 367 ACD395(4), ACD396(4), ACD397(4), ACD398(4), ACD399(4), ACD400(4),
C 371 ACD401(4), ACD402(4), ACD403(4), ACD404(4), ACD405(4), ACD406(4),
C 375 ACD407(4), ACD408(4), ACD409(4), ACD410(4), ACD411(4), ACD412(4),
C 379 ACD413(4), ACD414(4), ACD415(4), ACD416(4), ACD417(4), ACD418(4),
C 383 ACD419(4), ACD420(4), ACD421(4), ACD422(4), ACD423(4), ACD424(4),
C 387 ACD425(4), ACD426(4), ACD427(4), ACD428(4), ACD429(4), ACD430(4),
C 391 ACD431(4), ACD432(4), ACD433(4), ACD434(4), ACD435(4), ACD436(4),
C 395 ACD437(4), ACD438(4), ACD439(4), ACD440(4), ACD441(4), ACD442(4),
C 399 ACD443(4), ACD444(4), ACD445(4), ACD446(4), ACD447(4), ACD448(4),
C 403 ACD449(4), ACD450(4), ACD451(4), ACD452(4), ACD453(4), ACD454(4),
C 407 ACD455(4), ACD456(4), ACD457(4), ACD458(4), ACD459(4), ACD460(4),
C 411 ACD461(4), ACD462(4), ACD463(4), ACD464(4), ACD465(4), ACD466(4),
C 415 ACD467(4), ACD468(4), ACD469(4), ACD470(4), ACD471(4), ACD472(4),
C 419 ACD473(4), ACD474(4), ACD475(4), ACD476(4), ACD477(4), ACD478(4),
C 423 ACD479(4), ACD480(4), ACD481(4), ACD482(4), ACD483(4), ACD484(4),
C 427 ACD485(4), ACD486(4), ACD487(4), ACD488(4), ACD489(4), ACD490(4),
C 431 ACD491(4), ACD492(4), ACD493(4), ACD494(4), ACD495(4), ACD496(4),
C 435 ACD497(4), ACD498(4), ACD499(4), ACD500(4), ACD501(4), ACD502(4),
C 439 ACD503(4), ACD504(4), ACD505(4), ACD506(4), ACD507(4), ACD508(4),
C 443 ACD509(4), ACD510(4), ACD511(4), ACD512(4), ACD513(4), ACD514(4),
C 447 ACD515(4), ACD516(4), ACD517(4), ACD518(4), ACD519(4), ACD520(4),
C 451 ACD521(4), ACD522(4), ACD523(4), ACD524(4), ACD525(4), ACD526(4),
C 455 ACD527(4), ACD528(4), ACD529(4), ACD530(4), ACD531(4), ACD532(4),
C 459 ACD533(4), ACD534(4), ACD535(4), ACD536(4), ACD537(4), ACD538(4),
C 463 ACD539(4), ACD540(4), ACD541(4), ACD542(4), ACD543(4), ACD544(4),
C 467 ACD545(4), ACD546(4), ACD547(4), ACD548(4), ACD549(4), ACD550(4),
C 471 ACD551(4), ACD552(4), ACD553(4), ACD554(4), ACD555(4), ACD556(4),
C 475 ACD557(4), ACD558(4), ACD559(4), ACD560(4), ACD561(4), ACD562(4),
C 479 ACD563(4), ACD564(4), ACD565(4), ACD566(4), ACD567(4), ACD568(4),
C 483 ACD569(4), ACD570(4), ACD571(4), ACD572(4), ACD573(4), ACD574(4),
C 487 ACD575(4), ACD576(4), ACD577(4), ACD578(4), ACD579(4), ACD580(4),
C 491 ACD581(4), ACD582(4), ACD583(4), ACD584(4), ACD585(4), ACD586(4),
C 495 ACD587(4), ACD588(4), ACD589(4), ACD590(4), ACD591(4), ACD592(4),
C 499 ACD593(4), ACD594(4), ACD595(4), ACD596(4), ACD597(4), ACD598(4),
C 503 ACD599(4), ACD600(4), ACD601(4), ACD602(4), ACD603(4), ACD604(4),
C 507 ACD605(4), ACD606(4), ACD607(4), ACD608(4), ACD609(4), ACD610(4),
C 511 ACD611(4), ACD612(4), ACD613(4), ACD614(4), ACD615(4), ACD616(4),
C 515 ACD617(4), ACD618(4), ACD619(4), ACD620(4), ACD621(4), ACD622(4),
C 519 ACD623(4), ACD624(4), ACD625(4), ACD626(4), ACD627(4), ACD628(4),
C 523 ACD629(4), ACD630(4), ACD631(4), ACD632(4), ACD633(4), ACD634(4),
C 527 ACD635(4), ACD636(4), ACD637(4), ACD638(4), ACD639(4), ACD640(4),
C 531 ACD641(4), ACD642(4), ACD643(4), ACD644(4), ACD645(4), ACD646(4),
C 535 ACD647(4), ACD648(4), ACD649(4), ACD650(4), ACD651(4), ACD652(4),
C 539 ACD653(4), ACD654(4), ACD655(4), ACD656(4), ACD657(4), ACD658(4),
C 543 ACD659(4), ACD660(4), ACD661(4), ACD662(4), ACD663(4), ACD664(4),
C 547 ACD665(4), ACD666(4), ACD667(4), ACD668(4), ACD669(4), ACD670(4),
C 551 ACD671(4), ACD672(4), ACD673(4), ACD674(4), ACD675(4), ACD676(4),
C 555 ACD677(4), ACD678(4), ACD679(4), ACD680(4), ACD681(4), ACD682(4),
C 559 ACD683(4), ACD684(4), ACD685(4), ACD686(4), ACD687(4), ACD688(4),
C 563 ACD689(4), ACD690(4), ACD691(4), ACD692(4), ACD693(4), ACD694(4),
C 567 ACD695(4), ACD696(4), ACD697(4), ACD698(4), ACD699(4), ACD700(4),
C 571 ACD701(4), ACD702(4), ACD703(4), ACD704(4), ACD705(4), ACD706(4),
C 575 ACD707(4), ACD708(4), ACD709(4), ACD710(4), ACD711(4), ACD712(4),
C 579 ACD713(4), ACD714(4), ACD715(4), ACD716(4), ACD717(4), ACD718(4),
C 583 ACD719(4), ACD720(4), ACD721(4), ACD722(4), ACD723(4), ACD724(4),
C 587 ACD725(4), ACD726(4), ACD727(4), ACD728(4), ACD729(4), ACD730(4),
C 591 ACD731(4), ACD732(4), ACD733(4), ACD734(4), ACD735(4), ACD736(4),
C 595 ACD737(4), ACD738(4), ACD739(4), ACD740(4), ACD741(4), ACD742(4),
C 599 ACD743(4), ACD744(4), ACD745(4), ACD746(4), ACD747(4), ACD748(4),
C 603 ACD749(4), ACD750(4), ACD751(4), ACD752(4), ACD753(4), ACD754(4),
C 607 ACD755(4), ACD756(4), ACD757(4), ACD758(4), ACD759(4), ACD760(4),
C 611 ACD761(4), ACD762(4), ACD763(4), ACD764(4), ACD765(4), ACD766(4),
C 615 ACD767(4), ACD768(4), ACD769(4), ACD770(4), ACD771(4), ACD772(4),
C 619 ACD773(4), ACD774(4), ACD775(4), ACD776(4), ACD777(4), ACD778(4),
C 623 ACD779(4), ACD780(4), ACD781(4), ACD782(4), ACD783(4), ACD784(4),
C 627 ACD785(4), ACD786(4), ACD787(4), ACD788(4), ACD789(4), ACD790(4),
C 631 ACD791(4), ACD792(4), ACD793(4), ACD794(4), ACD795(4), ACD796(4),
C 635 ACD797(4), ACD798(4), ACD799(4), ACD800(4), ACD801(4), ACD802(4),
C 639 ACD803(4), ACD804(4), ACD805(4), ACD806(4), ACD807(4), ACD808(4),
C 643 ACD809(4), ACD810(4), ACD811(4), ACD812(4), ACD813(4), ACD814(4),
C 647 ACD815(4), ACD816(4), ACD817(4), ACD818(4), ACD819(4), ACD820(4),
C 651 ACD821(4), ACD822(4), ACD823(4), ACD824(4), ACD825(4), ACD826(4),
C 655 ACD827(4), ACD828(4), ACD829(4), ACD830(4), ACD831(4), ACD832(4),
C 659 ACD833(4), ACD834(4), ACD835(4), ACD836(4), ACD837(4), ACD838(4),
C 663 ACD839(4), ACD840(4), ACD841(4), ACD842(4), ACD843(4), ACD844(4),
C 667 ACD845(4), ACD846(4), ACD847(4), ACD848(4), ACD849(4), ACD850(4),
C 671 ACD851(4), ACD852(4), ACD853(4), ACD854(4), ACD855(4), ACD856(4),
C 675 ACD857(4), ACD858(4), ACD859(4), ACD860(4), ACD861(4), ACD862(4),
C 679 ACD863(4), ACD864(4), ACD865(4), ACD866(4), ACD867(4), ACD868(4),
C 683 ACD869(4), ACD870(4), ACD871(4), ACD872(4), ACD873(4), ACD874(4),
C 687 ACD875(4), ACD876(4), ACD877(4), ACD878(4), ACD879(4), ACD880(4),
C 691 ACD881(4), ACD882(4), ACD883(4), ACD884(4), ACD885(4), ACD886(4),
C 695 ACD887(4), ACD888(4), ACD889(4), ACD890(4), ACD891(4), ACD892(4),
C 699 ACD893(4), ACD894(4), ACD895(4), ACD896(4), ACD897(4), ACD898(4),
C 703 ACD899(4), ACD900(4), ACD901(4), ACD902(4), ACD903(4), ACD904(4),
C 707 ACD905(4), ACD906(4), ACD907(4), ACD908(4), ACD909(4), ACD910(4),
C 711 ACD911(4), ACD912(4), ACD913(4), ACD914(4), ACD915(4), ACD916(4),
C 715 ACD917(4), ACD918(4), ACD919(4), ACD920(4), ACD921(4), ACD922(4),
C 719 ACD923(4), ACD924(4), ACD925(4), ACD926(4), ACD927(4), ACD928(4),
C 723 ACD929(4), ACD930(4), ACD931(4), ACD932(4), ACD933(4), ACD934(4),
C 727 ACD935(4), ACD936(4), ACD937(4), ACD938(4), ACD939(4), ACD940(4),
C 731 ACD941(4), ACD942(4), ACD943(4), ACD944(4), ACD945(4), ACD946(4),
C 735 ACD947(4), ACD948(4), ACD949(4), ACD950(4), ACD951(4), ACD952(4),
C 739 ACD953(4), ACD954(4), ACD955(4), ACD956(4), ACD957(4), ACD958(4),
C 743 ACD959(4), ACD960(4), ACD961(4), ACD962(4), ACD963(4), ACD964(4),
C 747 ACD965(4), ACD966(4), ACD967(4), ACD968(4), ACD969(4), ACD970(4),
C 751 ACD971(4), ACD972(4), ACD973(4), ACD974(4), ACD975(4), ACD976(4),
C 755 ACD977(4), ACD978(4), ACD979(4), ACD980(4), ACD981(4), ACD982(4),
C 759 ACD983(4), ACD984(4), ACD985(4), ACD986(4), ACD987(4), ACD988(4),
C 763 ACD989(4), ACD990(4), ACD991(4), ACD992(4), ACD993(4), ACD994(4),
C 767 ACD995(4), ACD996(4), ACD997(4), ACD998(4), ACD999(4), ACD1000(4),
C 771 ACD1001(4), ACD1002(4), ACD1003(4), ACD1004(4), ACD1005(4), ACD1006(4),
C 775 ACD1007(4), ACD1008(4), ACD1009(4), ACD1010(4), ACD1011(4), ACD1012(4),
C 779 ACD1013(4), ACD1014(4), ACD1015(4), ACD1016(4), ACD1017(4), ACD1018(4),
C 783 ACD1019(4), ACD1020(4), ACD1021(4), ACD1022(4), ACD1023(4), ACD1024(4),
C 787 ACD1025(4), ACD1026(4), ACD1027(4), ACD1028(4), ACD1029(4), ACD1030(4),
C 791 ACD1031(4), ACD1032(4), ACD1033(4), ACD1034(4), ACD1035(4), ACD1036(4),
C 795 ACD1037(4), ACD1038(4), ACD1039(4), ACD1040(4), ACD1041(4), ACD1042(4),
C 799 ACD1043(4), ACD1044(4), ACD1045(4), ACD1046(4), ACD1047(4), ACD1048(4),
C 803 ACD1049(4), ACD1050(4), ACD1051(4), ACD1052(4), ACD1053(4), ACD1054(4),
C 807 ACD1055(4), ACD1056(4), ACD1057(4), ACD1058(4), ACD1059(4), ACD1060(4),
C 811 ACD1061(4), ACD1062(4), ACD1063(4), ACD1064(4), ACD1065(4), ACD1066(4),
C 815 ACD1067(4), ACD1068(4), ACD1069(4), ACD1070(4), ACD1071(4), ACD1072(4),
C 819 ACD1073(4), ACD1074(4), ACD1075(4), ACD1076(4), ACD1077(4), ACD1078(4),
C 823 ACD1079(4), ACD1080(4), ACD1081(4), ACD1082(4), ACD1083(4), ACD1084(4),
C 827 ACD1085(4), ACD1086(4), ACD1087(4), ACD1088(4), ACD1089(4), ACD1090(4),
C 831 ACD1091(4), ACD1092(4), ACD1093(4), ACD1094(4), ACD1095(4), ACD1096(4),
C 835 ACD1097(4), ACD1098(4), ACD1099(4), ACD1100(4), ACD1101(4), ACD1102(4),
C 839 ACD1103(4), ACD1104(4), ACD1105(4), ACD1106(4), ACD1107(4), ACD1108(4),
C 843 ACD1109(4), ACD1110(4), ACD1111(4), ACD1112(4), ACD1113(4), ACD1114(4),
C 847 ACD1115(4), ACD1116(4), ACD1117(4), ACD1118(4), ACD1119(4), ACD1120(4),
C 851 ACD1121(4), ACD1122(4), ACD1123(4), ACD1124(4), ACD1125(4), ACD1126(4),
C 855 ACD1127(4), ACD1128(4), ACD1129(4), ACD1130(4), ACD1131(4), ACD1132(4),
C 859 ACD1133(4), ACD1134(4), ACD1135(4), ACD1136(4), ACD1137(4), ACD1138(4),
C 863 ACD1139(4), ACD1140(4), ACD1141(4), ACD1142(4), ACD1143(4), ACD1144(4),
C 867 ACD1145(4), ACD1146(4), ACD1147(4), ACD1148(4), ACD1149(4), ACD1150(4),
C 871 ACD1151(4), ACD1152(4), ACD1153(4), ACD1154(4), ACD1155(4), ACD1156(4),
C 875 ACD1157(4), ACD1158(4), ACD1159(4), ACD1160(4), ACD1161(4), ACD1162(4),
C 879 ACD1163(4), ACD1164(4), ACD1165(4), ACD1166(4), ACD1167(4), ACD1168(4),
C 883 ACD1169(4), ACD1170(4), ACD1171(4), ACD1172(4), ACD1173(4), ACD1174(4),
C 887 ACD1175(4), ACD1176(4), ACD1177(4), ACD1178(4), ACD1179(4), ACD1180(4),
C 891 ACD1181(4), ACD1182(4), ACD1183(4), ACD1184(4), ACD1185(4), ACD1186(4),
C 895 ACD1187(4), ACD1188(4), ACD1189(4), ACD1190(4), ACD1191(4), ACD1192(4),
C 899 ACD1193(4), ACD1194(4), ACD1195(4), ACD1196(4), ACD1197(4), ACD1198(4),
C 903 ACD1199(4), ACD1200(4), ACD1201(4), ACD1202(4), ACD1203(4), ACD1204(4),
C 907 ACD1205(4), ACD1206(4), ACD1207(4), ACD1208(4), ACD1209(4), ACD1210(4),
C 911 ACD1211(4), ACD1212(4), ACD1213(4), ACD1214(4), ACD1215(4), ACD1216(4),
C 915 ACD1217(4), ACD1218(4), ACD1219(4), ACD1220(4), ACD1221(4), ACD1222(4),
C 919 ACD1223(4), ACD1224(4), ACD1225(4), ACD1226(4), ACD1227(4), ACD1228(4),
C 923 ACD1229(4), ACD1230(4), ACD1231(4), ACD1232(4), ACD1233(4), ACD1234(4),
C 927 ACD1235(4), ACD1236(4), ACD1237(4), ACD1238(4), ACD1239(4), ACD1240(4),
C 931 ACD1241(4), ACD1242(4), ACD1243(4), ACD1244(4), ACD1245(4), ACD1246(4),
C 935 ACD1247(4), ACD1248(4), ACD1249(4), ACD1250(4), ACD1251(4), ACD1252(4),
C 939 ACD1253(4), ACD1254(4), ACD1255(4), ACD1256(4), ACD1257(4), ACD1258(4),
C 943 ACD1259(4), ACD1260(4), ACD1261(4), ACD1262(4), ACD1263(4), ACD1264(4),
C 947 ACD1265(4), ACD1266(4), ACD1267(4), ACD1268(4), ACD1269(4), ACD1270(4),
C 951 ACD1271(4), ACD1272(4), ACD1273(4), ACD1274(4), ACD1275(4), ACD1276(4),
C 955 ACD1277(4), ACD1278(4), ACD1279(4), ACD1280(4), ACD1281(4), ACD1282(4),
C 959 ACD1283(4), ACD1284(4), ACD1285(4), ACD1286(4), ACD1287(4), ACD1288(4),
C 963 ACD1289(4), ACD1290(4), ACD1291(4), ACD1292(4), ACD1293(4), ACD1294(4),
C 967 ACD1295(4), ACD1296(4), ACD1297(4), ACD1298(4), ACD1299(4), ACD1300(4),
C 971 ACD1301(4), ACD1302(4), ACD1303(4), ACD1304(4), ACD1305(4), ACD1306(4),
C 975 ACD1307(4), ACD1308(4), ACD1309(4), ACD1310(4), ACD1311(4), ACD1312(4),
C 979 ACD1313(4), ACD1314(4), ACD1315(4), ACD1316(4), ACD1317(4), ACD1318(4),
C 983 ACD1319(4), ACD1320(4), ACD1321(4), ACD1322(4), ACD1323(4), ACD1324(4),
C 987 ACD1325(
```

FORTAN IV PLUS 702-51
 010907.FTN /TRISLOCKS/WR 10110125 29-JUL-78 PAGE 2

0031 -RITE(6,300)(TEMP(K),K01,PA1)
 0032 RETURN
 0033 200 F0RNAT11H,15X,'BIAS CORRECTION ',11X,'FOR ',12,' CATEGORIES')
 0034 300 F0RNAT11H,2615X,'CATEGORY LABEL ',11X,' POPULATION ',261777.
 1 10X,'UNCORRECTED ',11X,' ',12X,' CORRECTED ',11X,' ',12X,
 2 ' VARIANCE ',2X,' ',261777)
 0035 END

~~3-27~~
 29

Figure 12.- Concluded.

3.3.6 SUBROUTINE CLUST7

3.3.6.1 Linkage

Subroutine CLUST7 is called one to four times by CCIT7. CLUST7 calls RITE7 for data output and calls the PRINTC entry to PRINT7 for an output message.

3.3.6.2 Interface

CLUST7 interfaces with CCIT7 via COMMON block BUF (see section 3.3.1.2.1) and passed parameter Q, with RITE7 via COMMON block CLUSTER (see section 3.3.1.2.4), and with PRINT7 via passing parameter RCNUM on call.

3.3.6.3 Input

There is no input to this subroutine.

3.3.6.4 Output

Subroutine CLUST7 has no output.

3.3.6.5 Storage

This subroutine requires 836 words of storage.

3.3.6.6 Description

CLUST7 processes the CCIT 'C' records to provide the total number of clusters and the identity of the analyst-labeled (type 1) dot used to name each cluster.

On the first record processed, CLUST7 decodes bytes 4 and 5 to obtain the total number of clusters (CNUM) and bytes 6 and 7 to obtain the number of clusters contained on the record [RCNUM(1-15)]. Then for each cluster, the 12 bytes representing the cluster name (6 bytes) and dot name (6 bytes) used in labeling the cluster are copied into the array CNAME. PRINT7 is called via entry PRINTC to print a message containing the parameter RCNUM. When CNUM sets of data are written into CNAME, CLUST7 calls subroutine RITE7 for output.

3.3.6.7 Flow Chart

The flow diagram for subroutine CLUST7 is given in figure 13.

3.3.6.8 Listing

The listing for this subroutine is given figure 14.

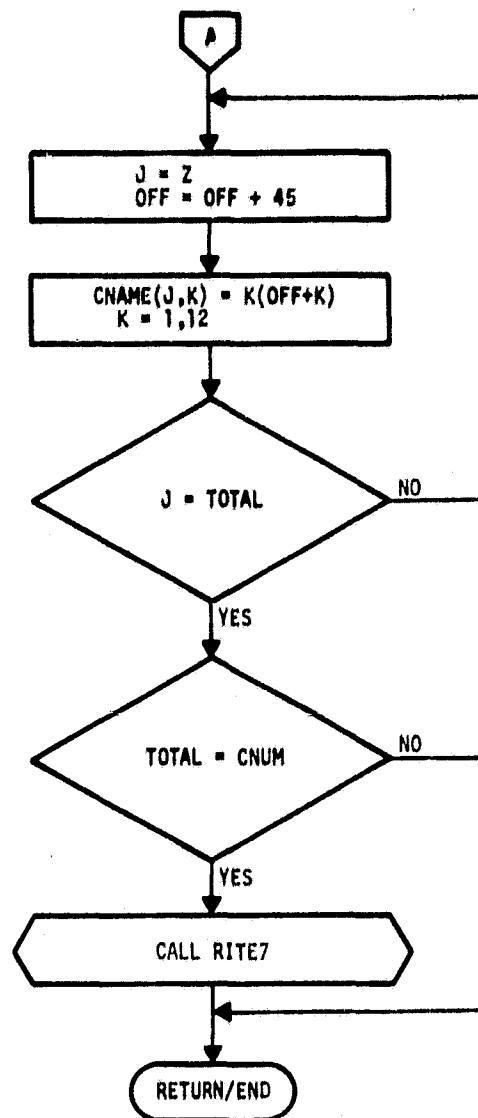
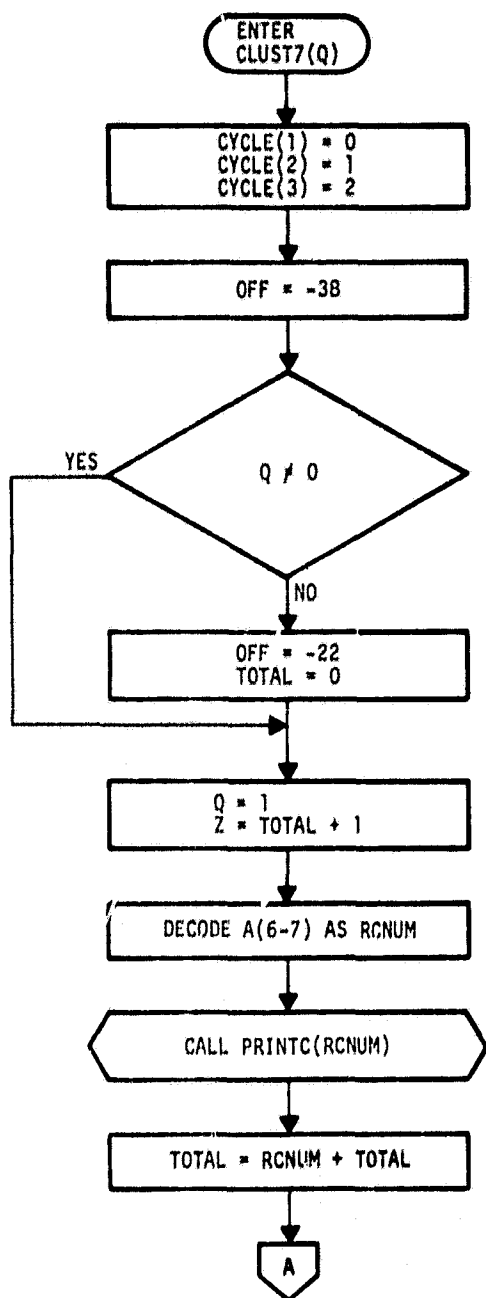


Figure 13.— Flow diagram for subroutine CLUST7.


```

FORTRAN IV-PLUS V02-51
CLUST7.FTN 14124148 06-JUL-78 PAGE 1
SUBROUTINE CLUST7(9)
IMPLICIT INTEGER(I02)
READS C01 'C' RECORDS AND OUTPUTS CLUSTER LABEL AND CLUSTER-
C DAY MATCH DATA TO DISC FILE NAMED SSSSYDDDD.CLO.
0003 C
C BYTE A(120), CYCLE(13), CNAME(1012)
C CNAME IS A STORAGE ARRAY TO PERMIT WRITING THE DATA IN ONE
C RECORD TO DISC IN A FILE NAMED SSSSYDDDD.CLO
C CNUM IS THE TOTAL NUMBER OF CLUSTERS
C CFF IS THE BYTE OFFSET WITHIN A 'C' RECORD
C CNUM IS THE NO. OF CLUSTERS IN THE CURRENT RECORD
C
C CNAME/BUFA
C CNAME/CLUSTER/CNAME,CNUM
C DATA CYCLE/100,141112/
0007 CFF = 538
0008 IF (CNAME.C) GO TO 5
0009 CFF = 22
0010 TOTAL = 0
0011 5 0 = 1
0012 2 = TOTAL + 1
0013 DECODE(2,100,AL0)CNUM
0014 CALL PRINT(CNUM)
0015 TOTAL = CNUM + TOTAL
0016 DO 1 J=0,TOTAL
0017 CFF = CFF + 45
0018 DO 1 K=1,12
0019 CNAME(K) = A(CFF+K)
C CHECK IF DONE
C CALL RITEEN FOR DISK WRITE AND PRINTED REPORT
C
C IF (TOTAL.C) CALL RITE7
C RETURN
0022 100
0023 END

```

Figure 14.— Listing for subroutine CLUST7.

3.3.7 SUBROUTINE RITE7

3.3.7.1 Linkage

Subroutine RITE7 is called once by subroutine CLUST7.

3.3.7.2 Interface

Subroutine RITE7 interfaces with CLUST7 via COMMON block CLUSTER (see section 3.3.1.2.4).

3.3.7.3 Input

There is no input to RITE7.

3.3.7.4 Output

RITE7 writes two records onto a previously opened file (unit 3). This unit is opened in subroutine BIASC7 as an unformatted FORTRAN disk file. In addition, RITE7 writes a line printer report of the cluster-dot match data for AA evaluation.

3.3.7.5 Storage

This subroutine requires 484 words of storage.

3.3.7.6 Description

RITE7 writes two records onto unit 3. The first record is a single integer, CNUM. The second record consists of the array CNAME as CNUM 12-byte elements. The output file is closed via a call to the system routine CLOSE.

3.3.7.7 Flow Chart

The flow diagram for subroutine RITE7 is given in figure 15.

3.3.7.8 Listing

The listing for this subroutine is given in figure 16.

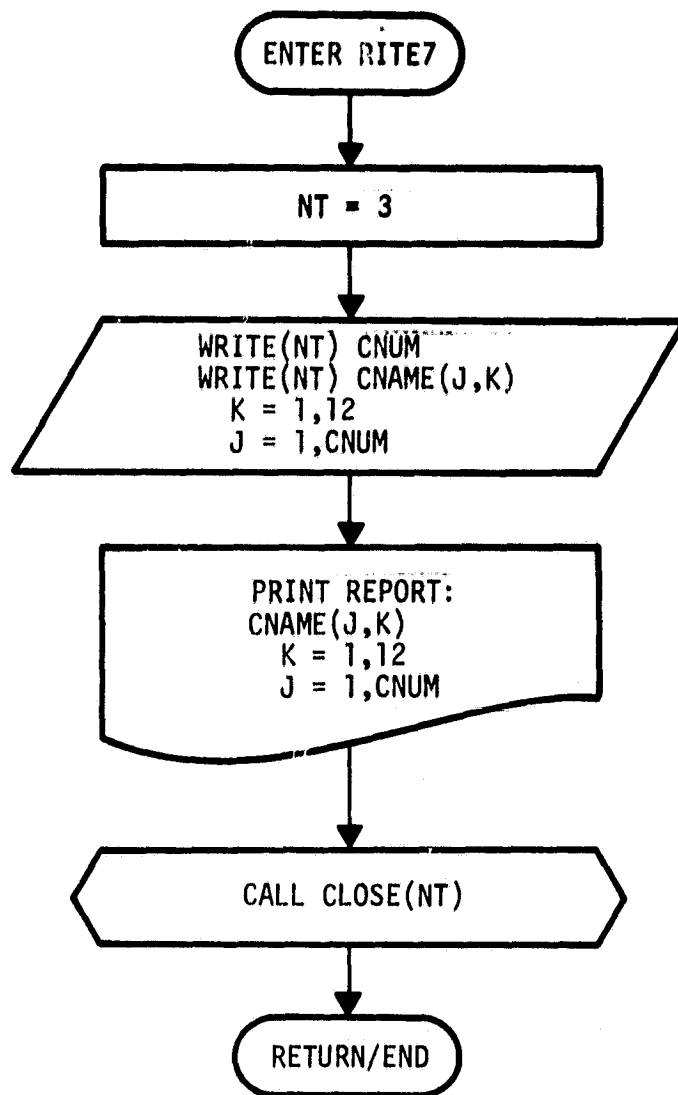


Figure 15.— Flow diagram for subroutine RITE7.

```

FBRDAY IV-PUS 10251 14112144 30-JUN-78 PAGE 1
RITE7.FIN

0001 SUPPLY THE RITE7
0002 IMPLICIT INTEGER(4*2)
C -WRITES AN UNFORMATTED FILE OF CLUSTER NAME AND CLUSTER # AT DBT MATCH
C DATA AND RISC FOR AA PROCESSING...LACIEY VERSION
C ALSO PRINTS REPORT FOR AA INTERNAL USE
C OUTPUT WAS ADDED IN SUBROUTINE BIASC
0003 BYTE CNAME(60,12)
0004 CNAME/CLUSTER/CNAME,CNUM
0005 N = 3
C -RITE 14P RECORDS, FIRST, THE NUMBER OF CLUSTERS, THEN, THE ALPHANUMERIC
C DATA, 12 CHARACTERS, PER CLUSTER
C -RITE(N)CNUM
0006 -RITE(N)((CNAME(J,K),K=1,12),J=1,CNUM)
0007 -RITE(N)((CNAME(J,K),K=1,12),J=1,CNUM)
0008 -RITE(6,100)((CNAME(J,K),K=1,12),J=1,CNUM)
C CLOSE OUTPUT FILE
0009 CALL CLOSEIN
0010 RETURN
0011 100 FBRDAY(14,15X,'CLUSTER NAME DBT NAME',//,60(18X,64X,9X,64X,7X))
0012 END

```

Figure 16.— Listing for subroutine RITE7.

ORIGINAL PAGE IS
OF POOR QUALITY

3.3.8 SUBROUTINE TURNON

3.3.8.1 Linkage

Subroutine TURNON is called by subroutines READH, BIASC7, and DOTS7.

3.3.8.2 Interface

TURNON interfaces with its calling routines via COMMON block NAME (see section 3.3.2.2.1) and a passed parameter, NT.

3.3.8.3 Input

There is no input to this subroutine.

3.3.8.4 Output

TURNON has no output.

3.3.8.5 Storage

This subroutine requires 162 words of storage.

3.3.8.6 Description

TURNON opens a file with the file name contained in byte array NM. If NT equals 1, the input file is opened as UNIT equals 1. If NT equals 2 to 6, an unformatted file is opened as unit NT. If NT is greater than 6, a formatted file is opened as unit (NT - 6). Prior to opening the file, the routine prints a message containing the passed unit number parameter, NT, and the file name, NM.

3.3.8.7 Flow Chart

The flow diagram for subroutine TURNON is given in figure 17.

3.3.8.8 Listing

The listing for this subroutine is given in figure 18.

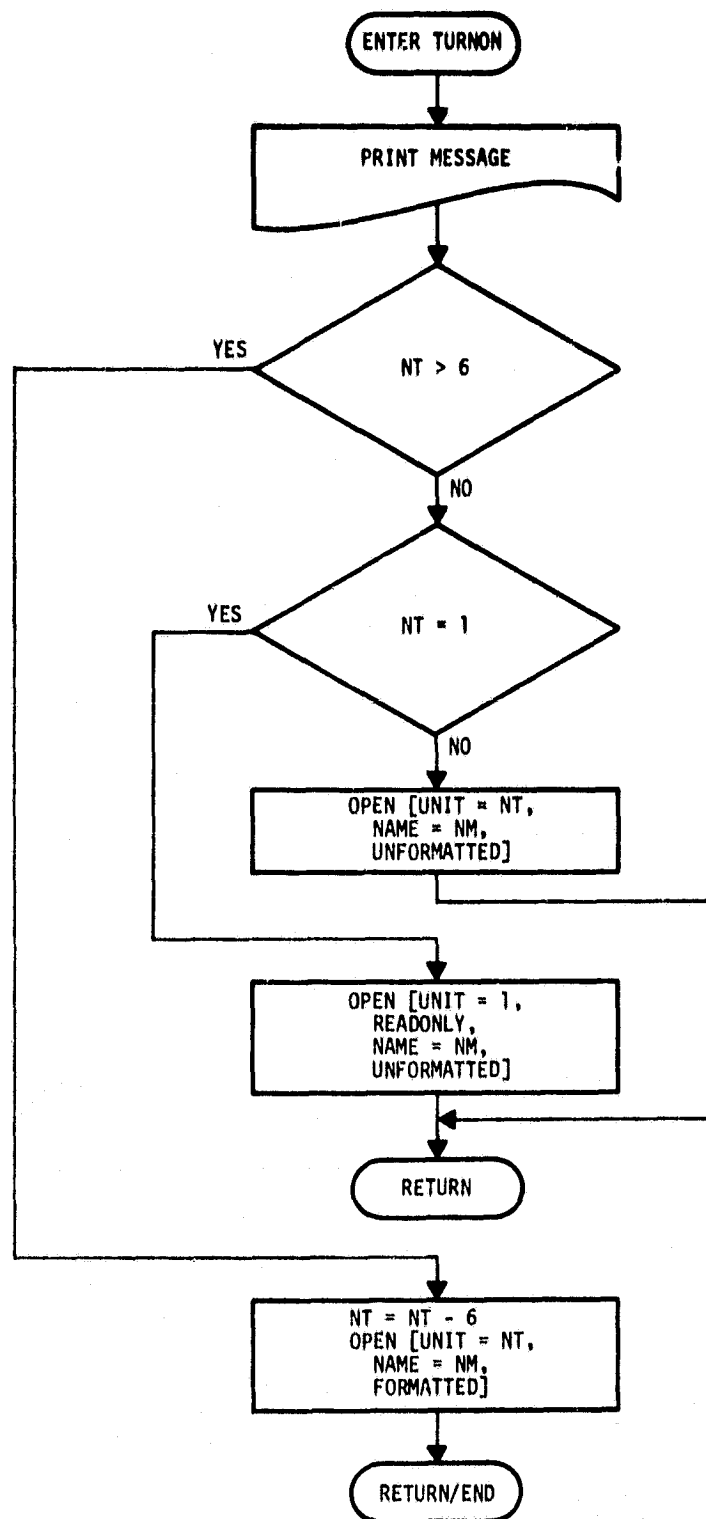


Figure 17.— Flow diagram for subroutine TURNON.

3.3.9 SUBROUTINE DOTS7

3.3.9.1 Linkage

Subroutine DOTS7 is called by CCIT7 once via the main entry and 14 times via entry DOTS1. DOTS7 calls subroutine TURNON twice and subroutines PRINT7 (via entry PRINTD) and STCOD7 once.

3.3.9.2 Interface

Subroutine DOTS7 interfaces with TURNON via COMMON block NAME (see section 3.3.2.2.1) and with CCIT7 via COMMON blocks BUF (see section 3.3.1.2.1) and DOTS (see section 3.3.4.2.1).

3.3.9.3 Input

There is no input to this subroutine.

3.3.9.4 Output

Subroutine DOTS7 writes formatted, card-image records onto two disk-based output files opened on the initial call to the routine.

3.3.9.5 Storage

This subroutine requires 976 words of storage.

3.3.9.6 Description

DOTS7 processes CCIT 'D' records into two formatted files of analyst-labeled dots. When called as DOTS7, the routine initializes the unit parameters, NT and MT, and the dot counters, KOUNT1 and KOUNT2. Then the elements of the array NM are set to name the file to receive the type 1 analyst-labeled dot data, and TURNON is called to open this file. NM(24) is redefined (1 → 2) to provide the name of the type 2 dot output file, and TURNON is called to open this file. Subroutine STCOD7 is called to obtain the two-byte parameter ST, the alphabetic state code for the segment. Control then returns to CCIT7.

When called as DOTS1, the routine processes one 720-byte 'D' record. For each analyst-labeled dot, one record is written. For type 1 dots, the data are written onto unit 2; for type 2 dots, the data are written onto unit 3. KOUNT1 is incremented for each type 1 dot, and KOUNT2 is incremented for each type 2 dot.

After processing all 209 dots (14 calls from CCIT7), a blank record is written into each output file. Then both output files are closed, and a message listing KOUNT1 and KOUNT2 is printed via a call to the PRINT7 subroutine entry PRINTD.

3.3.9.7 Flow Chart

The flow diagram for subroutine DOTS7 is given in figure 19.

3.3.9.8 Listing

The listing for this subroutine is given in figure 20.

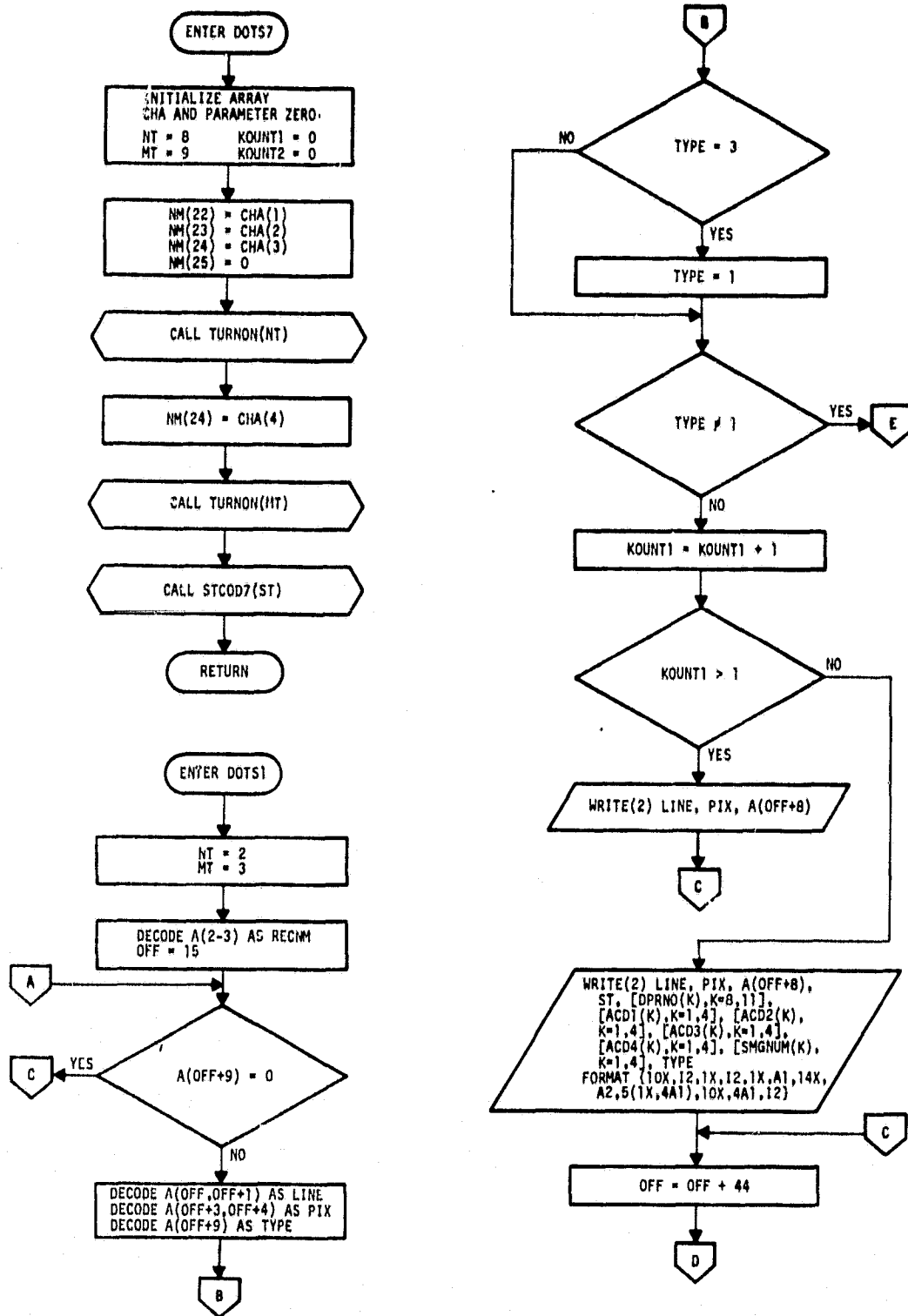


Figure 19.— Flow diagram for subroutine DOTS7.

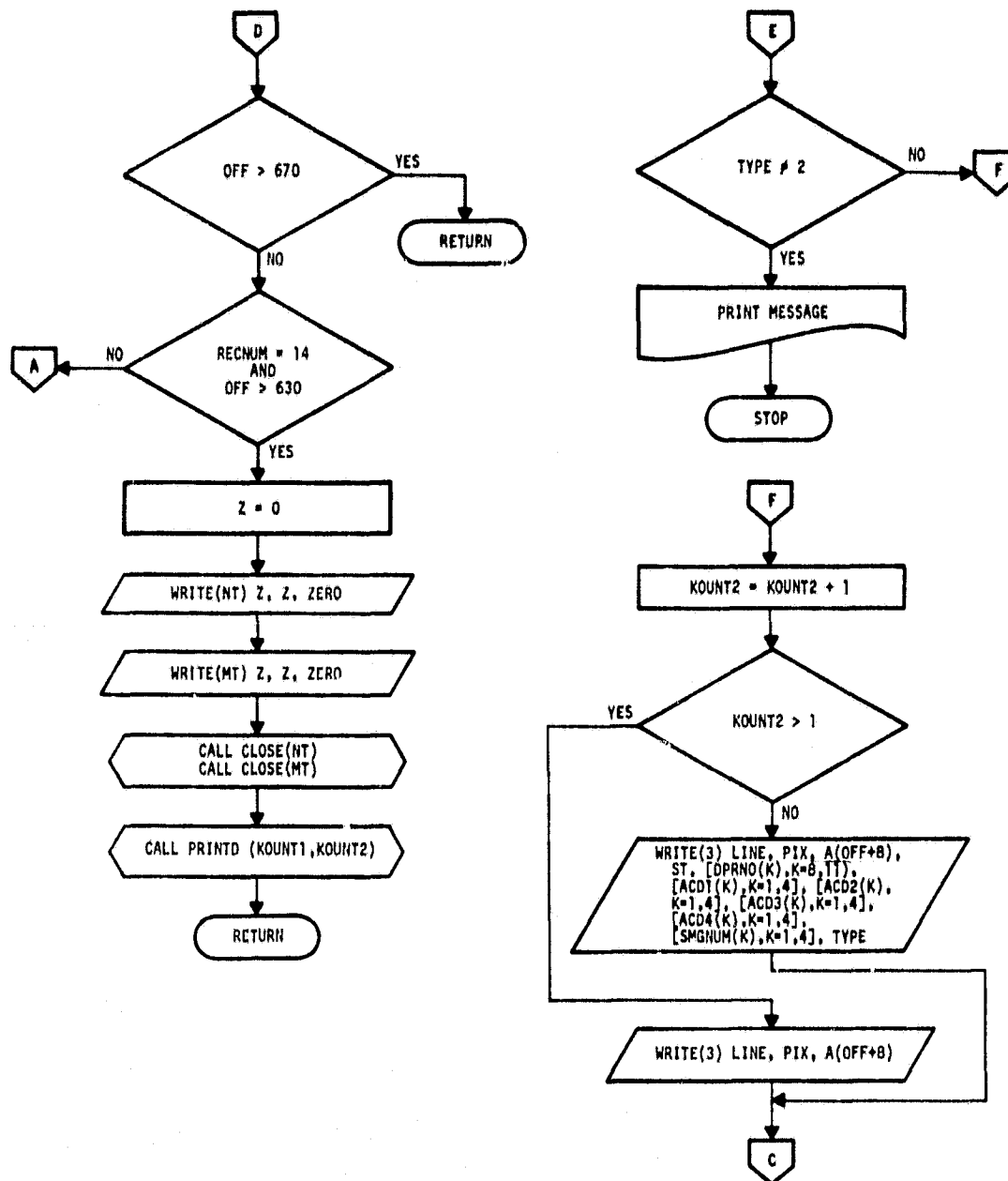


Figure 19.— Concluded.

3-41
43

Figure 20.— Listing for subroutine DOTS7.

ORIGINAL PAGE IS
OF POOR QUALITY

ADDRESS	DATA
0000	TYPE 1 DATA LABEL AS TYPE 1
0001	IF (TYPE, 1, 1) GOTO 100
0002	IF (TYPE, 2, 2) GOTO 200
0003	IF (TYPE, 3, 3) GOTO 300
0004	IF (TYPE, 4, 4) GOTO 400
0005	IF (TYPE, 5, 5) GOTO 500
0006	IF (TYPE, 6, 6) GOTO 600
0007	IF (TYPE, 7, 7) GOTO 700
0008	IF (TYPE, 8, 8) GOTO 800
0009	IF (TYPE, 9, 9) GOTO 900
0010	IF (TYPE, 10, 10) GOTO 1000
0011	IF (TYPE, 11, 11) GOTO 1100
0012	IF (TYPE, 12, 12) GOTO 1200
0013	IF (TYPE, 13, 13) GOTO 1300
0014	IF (TYPE, 14, 14) GOTO 1400
0015	IF (TYPE, 15, 15) GOTO 1500
0016	IF (TYPE, 16, 16) GOTO 1600
0017	IF (TYPE, 17, 17) GOTO 1700
0018	IF (TYPE, 18, 18) GOTO 1800
0019	IF (TYPE, 19, 19) GOTO 1900
0020	IF (TYPE, 20, 20) GOTO 2000
0021	IF (TYPE, 21, 21) GOTO 2100
0022	IF (TYPE, 22, 22) GOTO 2200
0023	IF (TYPE, 23, 23) GOTO 2300
0024	IF (TYPE, 24, 24) GOTO 2400
0025	IF (TYPE, 25, 25) GOTO 2500
0026	IF (TYPE, 26, 26) GOTO 2600
0027	IF (TYPE, 27, 27) GOTO 2700
0028	IF (TYPE, 28, 28) GOTO 2800
0029	IF (TYPE, 29, 29) GOTO 2900
0030	IF (TYPE, 30, 30) GOTO 3000
0031	IF (TYPE, 31, 31) GOTO 3100
0032	IF (TYPE, 32, 32) GOTO 3200
0033	IF (TYPE, 33, 33) GOTO 3300
0034	IF (TYPE, 34, 34) GOTO 3400
0035	IF (TYPE, 35, 35) GOTO 3500
0036	IF (TYPE, 36, 36) GOTO 3600
0037	IF (TYPE, 37, 37) GOTO 3700
0038	IF (TYPE, 38, 38) GOTO 3800
0039	IF (TYPE, 39, 39) GOTO 3900
0040	IF (TYPE, 40, 40) GOTO 4000
0041	IF (TYPE, 41, 41) GOTO 4100
0042	IF (TYPE, 42, 42) GOTO 4200
0043	IF (TYPE, 43, 43) GOTO 4300
0044	IF (TYPE, 44, 44) GOTO 4400
0045	IF (TYPE, 45, 45) GOTO 4500
0046	IF (TYPE, 46, 46) GOTO 4600
0047	IF (TYPE, 47, 47) GOTO 4700
0048	IF (TYPE, 48, 48) GOTO 4800
0049	IF (TYPE, 49, 49) GOTO 4900
0050	IF (TYPE, 50, 50) GOTO 5000
0051	IF (TYPE, 51, 51) GOTO 5100
0052	IF (TYPE, 52, 52) GOTO 5200
0053	IF (TYPE, 53, 53) GOTO 5300
0054	IF (TYPE, 54, 54) GOTO 5400
0055	IF (TYPE, 55, 55) GOTO 5500
0056	IF (TYPE, 56, 56) GOTO 5600
0057	IF (TYPE, 57, 57) GOTO 5700
0058	IF (TYPE, 58, 58) GOTO 5800
0059	IF (TYPE, 59, 59) GOTO 5900
0060	IF (TYPE, 60, 60) GOTO 6000
0061	IF (TYPE, 61, 61) GOTO 6100
0062	IF (TYPE, 62, 62) GOTO 6200

Figure 20.— Concluded.

3.3.10 SUBROUTINE STCOD7

3.3.10.1 Linkage

STCOD7 is called once by subroutine DOTS7.

3.3.10.2 Interface

STCOD7 interfaces with DOTS7 via COMMON block FNAME (see section 3.3.1.2.2) and passed parameter ST.

3.3.10.3 Input

There is no input to this subroutine.

3.3.10.4 Output

Subroutine STCOD7 has no output.

3.3.10.5 Storage

This subroutine requires 1099 words of storage.

3.3.10.6 Description

STCOD7 locates the correct two-character alphabetic state code, ST, for a given segment number via table lookup. Note: The table given is only valid for AA LACIE U.S. Great Plains blind sites for the 1978 Transition Year.

3.3.10.7 Flow Chart

The flow diagram for subroutine STCOD7 is given in figure 21.

3.3.10.8 Listing

The listing for this subroutine is given in figure 22.

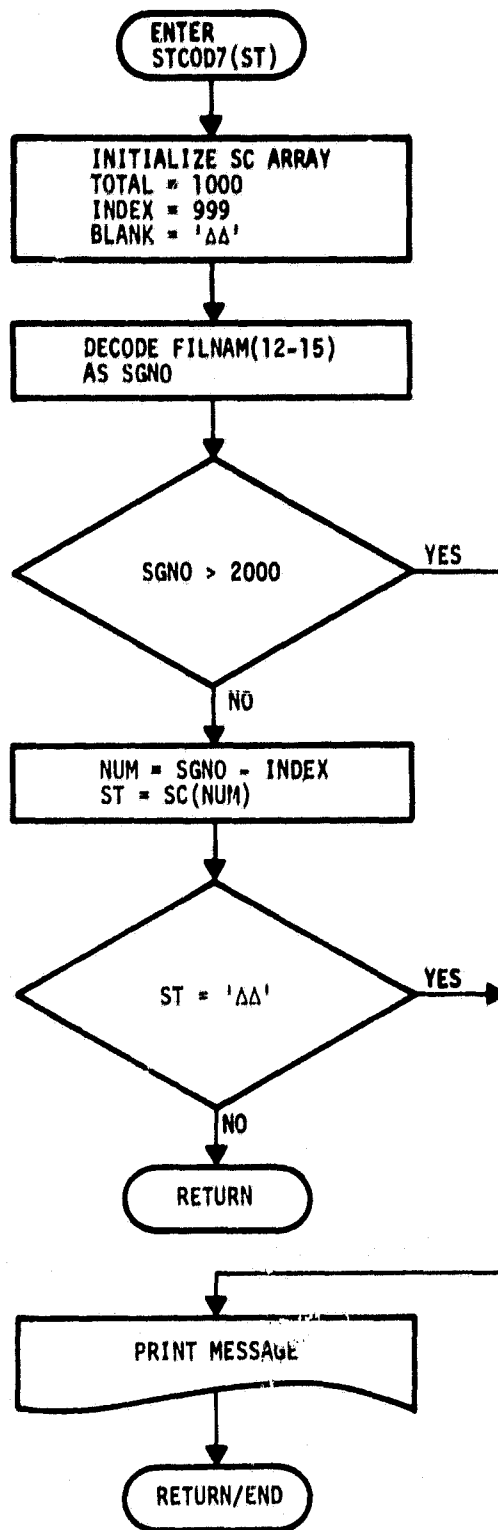


Figure 21.— Flow diagram for subroutine STCOD7.

3.3.11 SUBROUTINE PRINT7

3.3.11.1 Linkage

PRINT7 is called by CCIT7 via entries PRINT7 and PRINTE, by HEADER via entry PRINTH, by DOTS via entry PRINTD, and by CLUST7 via entry PRINTC. All other called routines are Image Processor system routines.

3.3.11.2 Interface

PRINT7 interfaces with HEADER via COMMON block DOTS (see section 3.3.4.2.1), with CCIT7 via COMMON block FNAME (see section 3.3.1.2.2), with DOTS7 via passed parameters K1 and K2, and with CLUST7 via passed parameter RENUM.

3.3.11.3 Input

There is no input to this subroutine.

3.3.11.4 Output

PRINT7 prints messages on the line printer.

3.3.11.5 Storage

This subroutine requires 783 words of storage.

3.3.11.6 Description

PRINT7 provides most line printer message output for the CCIT7 processor. This output provides a processing record for AA status and tracking activity. The routine uses system routines TIME and DATE to obtain data for header and trailer line printer messages for each run.

3.3.11.7 Flow Chart

The flow diagram for subroutine PRINT7 is given in figure 23.

3.3.11.8 Listing

The program listing for this subroutine is given figure 24.

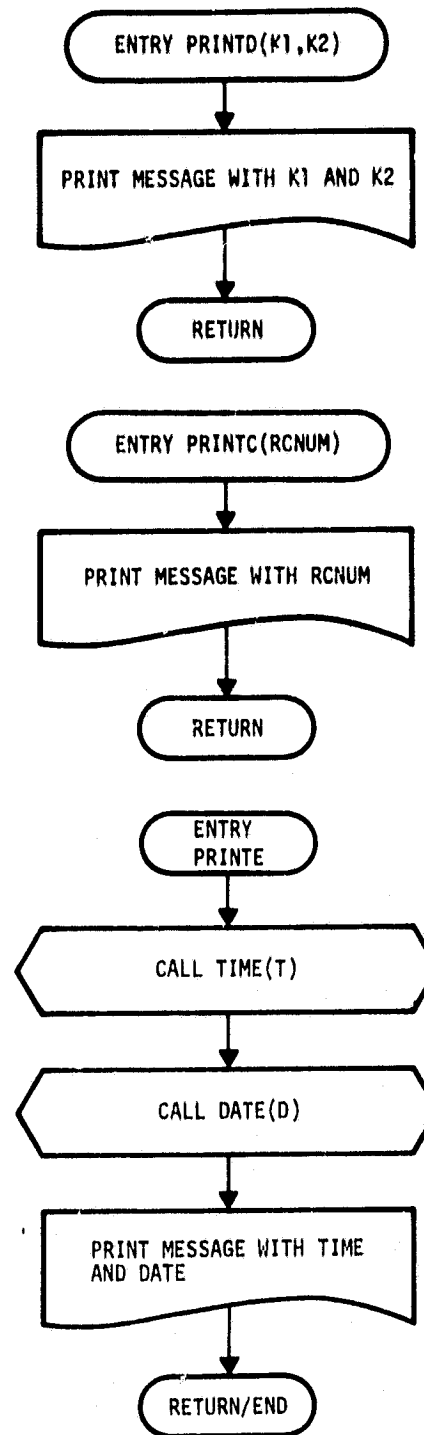
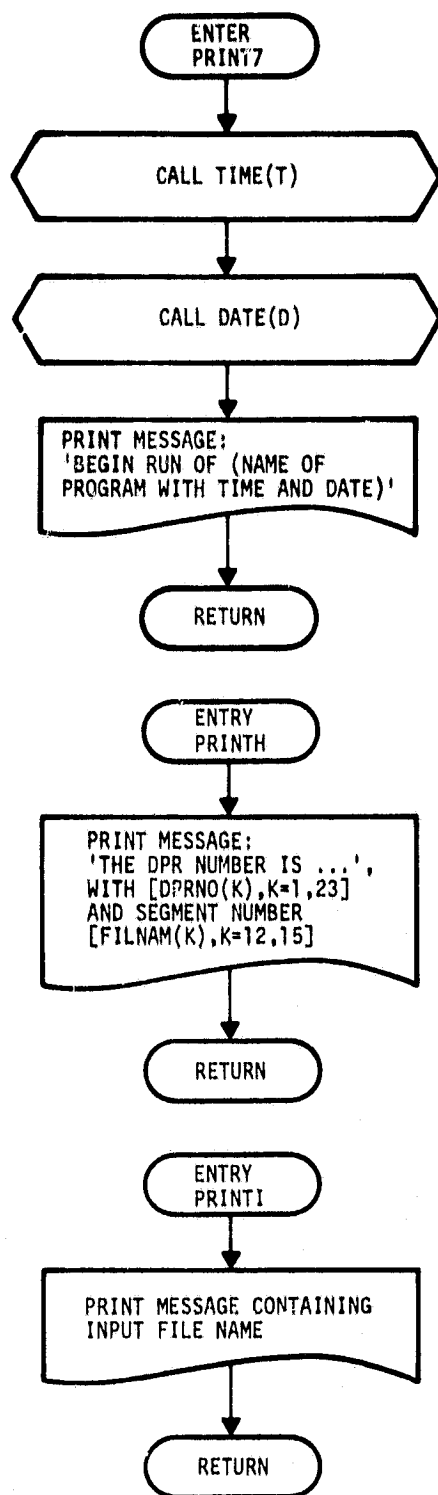


Figure 23.— Flow diagram for subroutine PRINT7.

4. OPERATIONS

This section presents all the information necessary to obtain proper execution of the CCIT7 processor program.

4.1 OPERATORS GUIDE

This section explains the system hardware configuration and execution (run) setup for the CCIT7.

4.1.1 HARDWARE CONFIGURATION

The nominal configuration is the Earth Observations Division/Data Techniques Laboratory (EOD/DTL) PDP 11/45 processor with the RSX 11-D operating system. The system must have the input CCIT files resident on either the system disk or a user disk. The output files are written onto the same disk and under the same user identification code (UIC) as the resident input data. The input files are created using program AACCIT, described in JSC-13893. (See section 2 of this specification.)

4.1.2 PROGRAM EXECUTION

4.1.2.1 INTERACTIVE SETUP

- a. Edit file CCIT7.DAT for the proper file name and the value of parameter SKIP (24A1,I2). The file name takes the form:

DBX:[abc,d]SSSSYYDDD.wxy

where

X = Disk unit number

SSSSYYDDD = Input file name

wxy = Input file type; i.e., .CCO

[abc,d] = UIC for the input file

41
520

- b. Mount the proper disk pack on the drive.
- c. Type 'RUN CCIT7'.
- d. When message CCIT7-STOP appears on the monitor, collect a single-page report at the line printer, and check the listing to ensure that the ending message was printed and that the various steps were properly executed.

4.1.2.2 BATCH SETUP

- a. Prepare a batch run request detailing the disk configuration required.
- b. Set up a batch run deck as in table 2. The required steps follow:
 - Delete CCIT7.DAT.
 - Create CCIT7.DAT with card images, as given in section 4.1.2.1.
 - Run CCIT7.TSK.

4.2 USERS GUIDE

The CCIT7 program is designed to obtain a small fraction of the data from a CCIT disk file and to reformat these data into a form directly used by several AA software modules. This program will not execute correctly for CCIT's other than those created under LACIE version 7. The approximate dates of valid CCIT's for version 7 are 77305 through the date of issuance of this specification.

4.3 MAINTENANCE DOCUMENTATION

Not applicable.

TABLE 2.— BATCH RUN DECK SETUP

\$JOB/NAME=AA/MCR/LIMIT=99/ACCOUNT=1106

\$MCR PIP

CCIT7.DAT;*/DE

\$CREATE CCIT7.DAT

:

Card images for file name and SKIP parameter (24A1,I2)

:

Blank card

\$EOD

\$MCR REM RSXBAT

\$RUN CCIT7.TSK

\$EOJ

APPENDIX
FORMAT OF .CLO FILE

APPENDIX
FORMAT OF .CLO FILE

Record 1: The first record contains a single integer representing the number of classes detailed in the next record. Sixteen bytes of data follow for each class (minimum of 2 classes, maximum of 26 classes).

Record 2:

Byte number	Data description (ASCII)
1	Class 1 label (W, S, G, N, etc.)
2-6	Pixel population; PPPPP
7-9	Uncorrected proportion; M.MM (implied decimal point)
10-12	Corrected proportion; N.NN (implied decimal point)
13-16	Variance; .VVVV (implied decimal point)
17	Class 2 label (W, S, G, N, etc.)
18-22	Pixel population; PPPPP
23-25	Uncorrected proportion; M.MM (implied decimal point)
26-28	Corrected proportion; N.NN (implied decimal point)
29-32	Variance; .VVVV (implied decimal point)
⋮	⋮

Record 3: This record contains a single integer giving the number of clusters in the classification, CNUM.

Record 4: The fourth record contains 12 bytes of ASCII character data for each cluster; e.g., 12*CNUM bytes of data. The first 6 bytes of each group of 12 are the cluster label; e.g., NOCL17. The last six bytes of each group are the identity of the dot used to label the clusters; e.g., DOT103. Only type 1 dots are used to label clusters.

A2
51